Verification Method of Conditional Probability Based on Automaton

Mingyu Ji [1, 2]

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

Di Wu

College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China

Zhiyuan Chen

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract—**According to the demand of credible property verification for complex information system, this paper presents a kind of until formula conditional probability property verification and analysis method acting on discrete probability model. A new more expressive probabilistic computation tree logic used to describe until formula conditional probability property of system model. We express until path formula as automaton and give the formal representation of until formula intersection operation automaton. The method of calculate the corresponding accept state reachable probability be described based on product model which realizes the simultaneous evolution of the model and c operation automaton. The example result verifies the feasibility and validity of the method.**

Index Terms—**model checking; until formula; conditional probability; intersection operation; automaton**

I. INTRODUCTION

Along with the fast development of information technology, a variety of software and hardware systems become more and more complex, the credibility of system is becoming more and more important. How to use effective means to verify the functionality and performance of system has become an important research topic [1]. Model checking [2], as one kind of verification techniques, has been concentrated widely because of its high automation and providing counterexample path. Currently, model checking has got a great success in verification of computer hardware, communication protocol [3], software system [4-6], etc. Traditional model checking technology utilizes the computation tree logic and linear temporal logic to represent properties to be verified, the both logics can specify absolute correctness of system behavior, such as system operation cannot fail, and so on.

In practice, however, there are a lot of random phenomena, such as message loss in unreliable channel. For this kind of phenomena, some probabilistic metric are often needed, such as failure probability of message transmission not higher than 1%, and so on. For such properties, computation tree logic and linear temporal logic cannot portray them, so probability operator is introduced based on the computation tree and linear temporal logic by researchers and the corresponding verification methods of probabilistic model are proposed.

Probabilistic model checking technology [7] expands traditional model checking theory, it can describe and verify the random property, time property and resource consumption property in system model comprehensively, which has been applied in practical problems, commonly in quantitative verification of complex property path formula. In the process of verification, finite Markov chain generally be adopted to describe system model and probabilistic computation tree logic formula [8] and continuous stochastic logic formula are generally used to describe the property of system. On these logic formula, multiple until formula can describe the periodic oscillations changes of biological species and other important characteristics in the field of systems biology, which makes high-performance verification and counterexample analysis about this kind of property become an open research topic. In the literature [9], stratified continuous time Markov chain model is used to explain the verification process of multiple until formula property with time boundaries, the corresponding verification algorithm and example analysis process are proposed by constructing product model of formula property automaton and continuous time Markov chain model.

In real world applications, as one of the basic concepts of probability theory, conditional probability is used to analyze and solve many problems, such as the risk assets assessment, anonymous protocol analysis and so on. The cases analysis of performance verification of conditional probability property has important practical significance. The basic representation form of conditional probability is $p[\varphi|\psi]=p[\varphi\wedge\psi]/p[\psi]$, where φ and ψ are logic formula to be verified.

Supported by the Fundamental Research Funds for the Central Universities under Grant No.DL11BB08, National College Students' Innovative projects under Grant No.201310225064 and the National Natural Science Foundation of China under Grant No. 71001023

doi:10.4304/jnw.8.6.1329-1335

The literature [10] describes issues of conditional probability property verification for continuous time Markov chain, and gives the approximate property verification algorithm and algorithm analysis. The literature [11] gives the model checking method of conditional probability computation tree logic counterexample for the discrete time Markov decision process.

This paper considers combining the multi until formula property with conditional probability, based on traditional qualitative model checking and automaton theory to study the quantitative verification method of multi until formula conditional probability property with complex constraints. The rest of paper is organized as follows: Section 2 gives the definition of the corresponding probability model. Section 3 introduces temporal logic syntax and gives a brief description of the conditional probability state formula semantics. Section 4 studies the automaton representation of property formula, introduces construction method of multi until formula intersection operation automaton, and gives the basic solution method of conditional probability. Section 5 gives construction method of automaton product model of particular form and describes the quantitative verification method. Section 6 gives verification process and calculation results based on constructed instance. The last section summarizes this paper and points out the research directions in the future.

II. Basic Discrete Probability Model

Probabilistic model has been widely used in various systems' design and analysis, there are different probabilistic models for different types of system features, wherein the Discrete Time Markov Chains (DTMC) is used to represent probability selection; Continuous Time Markov Chain (CTMC) is used to represent continuous time and probability selection; Markov decision process adds uncertainty based on DTMC, which is used to represent the probability and uncertainty selection.

This paper requires to use discrete probability model with the ability of resource feature description to model the system, but the models above do not have the ability in this regard, so first this paper describes the representation methods of object model to be studied, puts forward the definition of Discrete Time Markov Rewards Model (DTMRM), which supports transition resource consumption, and then gives instance specification of model representation.

Definition 1 DTMRM.

DTMRM is sextuple M = (S, P, L, AP, N, v), wherein S is a finite set of states, $P: S\times S \to [0,1]$ is the state transition probability matrix and for all states s, $\sum_{s'\in S} p(s,s')=1$, $L: S \to 2^{\mathrm{AP}}$ is the state labeling function, AP represents finite set of atomic proposition, $N: S\times S \to R_{\geq 0}$ represents transition consumption structure, $v \in Distr(S)$ for the sets of initial distribution.

Example 1 DTMRM example

Fig. 1 is a DTMRM containing six states and nine transitions, wherein s_0 is initial state, state s_1 satisfies the atomic propositions set {b, c, d}, there are four transitions in the system model transition process, and accompanying with probability selection and resource consumption in the transition process. $P(s_1,s_2)=0.5$ and $N(s_1,s_2)=20$ represent that there exit a state transition between s_1 and s_2 , transition probability is 0.5 and transition resource consumption is 20 in this transition process.

Resources herein primarily refers generic name of all non-temporality resources that the system interacts with operating environment, such as storage space, network bandwidth, power consumption of energy or physical space [12] and other physical resources occupied or consumed, as well as the negotiation procedure, active components and other logic resources.

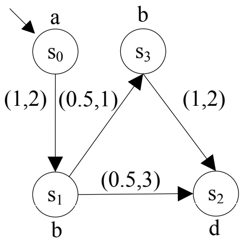


Figure 1. DTMRM example

III. Probability Computation Tree Logic and Conditional Probability

In probabilistic model checking process, the property to be verified of discrete probability model is generally described by Probabilistic Computation Tree Logic (PCTL). PCTL introduces probabilistic operator on the basis of Computation Tree logic (CTL), and can achieve quantitative verification analysis of model property. But PCTL has the limitation that any temporal operator must follow a state formula, which limits its expression relatively and does not have the ability to describe the transition resource consumption and the conditional probability formula in probability model transition process, so the paper extends PCTL, introduces transition resource consumption and conditional probability state formula, and puts forward Conditional Probability Probabilistic Reward Computation Tree Logic (CPPRCTL).

Definition 2 CPPRCTL syntax.

CPPRCTL state formula:

$$\Phi ::= true \big| f \big| \Phi_1 \wedge \Phi_2 \big| \neg\Phi \big| P_{\Delta p}(\varphi)$$

CPPRCTL path formula:

$$\varphi ::= \Phi \big| \varphi_1 \wedge \varphi_2 \big| \varphi_1 \big| \varphi_2 \big| \neg\varphi \big| \bigcirc_r^n \varphi \big| \varphi_1 \cup_r^n \varphi_2$$

where n=$[n_l,n_h] \subseteq R_{\geq 0}$ and r=$[r_l,\mathrm{r_h}] \subseteq R_{\geq 0}$ be transition step interval and transition consumption interval, Δ be arithmetic comparison operators, $\bigcirc$ be next operator, $\cup$ be until operator, f be atomic proposition, $p \in [0,1]$ be the probability value interval. One path satisfies path formula $\Phi_1 \bigcup_{\le r_1}^{\le n_1} \Phi_2$ means that formula Φ_2 in this path is satisfied in some state ultimately, and all states in the path before that state satisfy formula Φ_1, and transition steps number of this path is less than or equal to n_1, the total resource consumption in transition process is less than or equal to r_1.

Conditional probability state formula $p_{\le a}[\varphi|\psi]$ holds if and only if paths set starting at the current state satisfies path formula ψ and the holding probability of path formula φ is less than or equal to *a*.

This paper mainly researches the quantitative verification methods that conditional formula $\varphi|\psi$ based on automaton theory, where φ and ψ are multi until path formula with the form of $\Phi_1 \bigcup_{r_1}^{n_1} \Phi_2 \bigcup_{r_2}^{n_2} \cdots \Phi_k$. For a state s in the model to be verified, the satisfying probability of conditional formula $\varphi|\psi$ is represented as $prob(s,\varphi|\psi)$, its value can be calculated by means of the following equivalent form:

$$prob(s,\varphi|\psi) \Leftrightarrow {prob(s,\varphi \wedge \psi)}/{prob(s,\psi)}$$

IV. Formula Automaton Description

For the representation of a variety of temporal logic properties, scholars have done a lot of work. The commonly used representation methods for logics include Petri net [13-14], automaton technology [15] and so on. Because of low computational space complexity as well as strong ability of expression and many other advantages, the logic representation based on automaton theory is widely used [16]. The literature [17] uses automaton to represent the linear temporal logic of activity diagram in the Unified Modeling Language (UML), then verifies the corresponding logic feature by model checking technology and achieves good results. The literature [18] describes web services based on extended finite automaton, does formal verification to composite Web services, reducing complexity of verification computation, improving the verification efficiency.

In this section, Finite Automaton (FA) is used to represent the CPPRCTL multi until formula, and researches the automaton model representation method of multi until formula intersection operation involving in conditional probability by the semantic analysis of accepted language set of FA, and lays foundation for the structure and algorithm design of subsequent product model.

In the following, we give the definition of Multi Until Formula Nondeterministic Finite Automaton (MUFNFA) by the semantic analysis of multi until formula.

Definition 3 FA

FA 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, $F \subseteq Q$ be terminated state set.

In FA, the transition function δ is divided into two forms: If the form of transition function is expressed as $\delta: Q \times \Sigma \to Q$ and the number of states in initial state set of FA is less than or equal to one, then the FA named Deterministic Finite Automaton (DFA). If the form of transition function is expressed as $\delta: Q \times \Sigma \to 2^Q$, then the FA named Nondeterministic Finite Automaton (NFA).

Definition 4 accepted language of FA

Let $B = (\Sigma, Q, Q_0, \delta, F)$ be a FA, $A_i \in \Sigma$ be input alphabet of FA, $q_i \in Q$ be a state in FA, then finite sequence of states $q_0 q_1 \ldots q_n$ is called accepting run of B if and only if $q_0 \in Q_0 \wedge q_n \in F$ and for all $0 \le i < n$, $q_i \xrightarrow{A_{i+1}} q_{i+1}$. Letters finite set of accepting run of B is called accepted words of B, denoted as $w = A_1 A_2 \ldots A_n \in \sum^*$. The accepted language of B, denoted by $L(B)$, is the set of finite words in $\sum^*$ accepted by B, i.e.:

$$L(B) = \{w \in \Sigma^* \,|\, \delta^*(q_0, w) \cap F \ne \varnothing \wedge q_0 \in Q_0\}.$$

Wherein $\delta^*(q_0, w)$ represents reachable state set of automaton corresponding input word w starting at q_0.

For DFA, δ^* denoted by $\delta^*: Q \times \Sigma^* \to Q$ and $\delta^*(q_0, w) = \delta^*(q_0, A_1 A_2 \ldots A_n) = \delta^*(\delta(q_0, A_1), A_2 \ldots A_n)$

For NFA, δ^* denoted by $\delta^*: Q \times \Sigma^* \to 2^Q$ and $\delta^*(q_0, w) = \delta^*(q_0, A_1 A_2 \ldots A_n) = \bigcup_{p \in \delta(q_0, A_1)} \delta^*(p, A_2 \ldots A_n)$

Definition 5 size of automaton

Let $B = (\Sigma, Q, Q_0, \delta, F)$ be a FA, then the size of B is expressed as $|B|$, its value is the sum of state number and transition number in B, i.e.:

$$|B| = |Q| + \sum_{q \in Q} \sum_{A \in \Sigma} |\delta(q, A)|$$

Example 2 instance of FA

An instance of NFA is shown in Fig. 2, where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{A_1, A_2\}$, $Q_0 = \{q_0\}$, $F = \{q_2\}$, and the transition function δ is defined by:

$\delta(q_0, A_1) = \{q_0\}$ $\delta(q_0, A_2) = \{q_0, q_1\}$

$\delta(q_1, A_1) = \{q_2\}$ $\delta(q_1, A_2) = \{q_2\}$

$\delta(q_2, A_1) = \{\varnothing\}$ $\delta(q_2, A_2) = \{\varnothing\}$

The accepted language of this automaton is given by the regular expression $(A_1 + A_2)^* A_2 (A_1 + A_2)$.

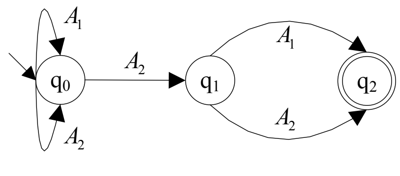


Figure 2. Instance of FA

Definition 6 total DFA

A DFA is called total DFA if and only if $|Q_0| = 1$ and for all $q \in Q$ and for all $A \in \Sigma$, $|\delta(q, A)| = 1$.

Definition 7 NFA Determination

Let $B=(\Sigma,Q,Q_0,\delta,F)$ be an NFA, its equivalent total DFA can be done by a power set construction, expressed as $B_{\det}=(\Sigma,2^Q,Q_0,\delta_{\det},F_{\det})$, where the termination state set is expressed as $F_{\det}=\{Q'\subseteq Q \mid Q'\cap F\neq\varnothing\}$ and the transition function $\delta_{\det}:2^Q\times\Sigma\rightarrow 2^Q$ is defined by: $\delta_{\det}(Q',A)=\bigcup_{q\in Q'}\delta(q,A)$.

Example 3 NFA total determined

The NFA is shown in Fig. 2, then the total DFA by power set construction be shown in Fig. 3.

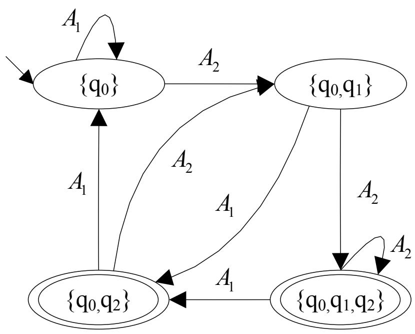


Figure 3. Total DFA

Definition 8 MUFNFA

For CPPRCTL path Formula $\varphi=f_1\cup f_2\cup\cdots\cup f_k$, 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 4 MUFNFA example

Fig. 4 is the corresponding MUFNFA of multi until formula $f_1\cup f_2\cup f_3$, where q_0 is the initial state and q_3 is terminate state.

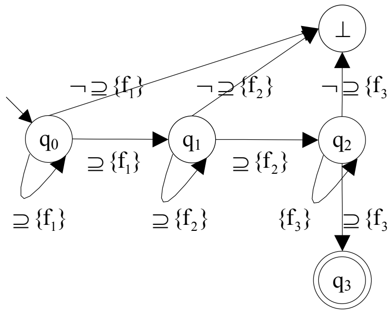


Figure 4. NFA of formula $f_1\cup f_2\cup f_3$

Definition 9 multi until formula intersection automaton

Let $B_{\varphi_i}=(\Sigma_i,Q_i,Q_{0i},\delta_i,F_i)$, where the symbol i is equal to 1 or 2, represent the corresponding MUFNFA of path formula $\varphi_i=\Phi_1\cup\Phi_2\cup\cdots\cup\Phi_k$, then the corresponding MUFNFA of intersection operation $\varphi_\cap=\varphi_1\wedge\varphi_2$ of φ_1 and φ_2, named as $B_\cap$, is expressed as:

$$\begin{aligned}B_\cap&=B_{\varphi_1}\otimes B_{\varphi_2}\\&=(\Sigma_\cap,Q_\cap,Q_{0\cap},\delta_\cap,F_\cap)\\&=(\Sigma_1\cup\Sigma_2,Q_1\times Q_2,Q_{01}\times Q_{02},\delta_\cap,F_1\times F_2)\end{aligned}$$

The representation of transition function $\delta_\cap$ is divided into three cases:

$$\delta_\cap=\frac{q_i\xrightarrow{A_1}{}_1\,q_j\wedge q_k'\xrightarrow{A_2}{}_2\,q_l'}{(q_i,q_k')\xrightarrow{A_1\cup A_2}(q_j,q_l')},$$

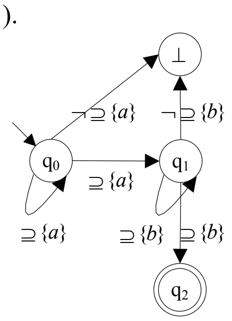
$$\delta_\cap=\frac{q_i\in F_1\wedge q_k'\xrightarrow{A_2}{}_2\,q_l'}{(q_i,q_k)\xrightarrow{A_2}(q_i,q_l')},\text{ and}$$

$$\delta_\cap=\frac{q_i\xrightarrow{A_1}{}_1\,q_j\wedge q_k\in F_2}{(q_i,q_k')\xrightarrow{A_1}(q_j,q_k')}.$$

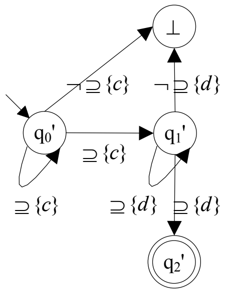
The accepted language of MUFNFA $B_\cap$ of intersection operation $\varphi_\cap=\varphi_1\wedge\varphi_2$ is expressed as $L(B_\cap)$, accepted language of $B_\cap$ is the intersection of the accepted language of B_{φ_1} and the accepted language of B_{φ_2}, denoted by $L(B_\cap)=L(B_{\varphi_1}\otimes B_{\varphi_2})=L(B_{\varphi_1})\cap L(B_{\varphi_2})$.

Example 5 MUFNFA accepted language intersection

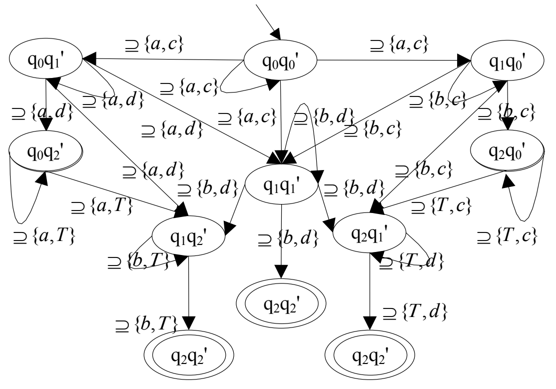
Fig. 5(1) and 5(2) show the MUFNFA of single until formula $a\cup b$ and $c\cup d$. The intersection operation MUFNFA of until formula $a\cup b$ and $c\cup d$ can be shown in Fig. 5(3) by definition 9, where q_0q_0' is the initial state, q_2q_2' is the terminate state. In order to simplify, the states containing $\perp$ and the related transitions are omitted in Fig. 5 (3).



(1) $a\cup b$



(2) $c\cup d$



(3) $a \bigcup b \bigcap c \bigcup d$

Figure 5. MUFNFA instance of until formula intersection operation

Due to $prob(s,\varphi|\psi) \Leftrightarrow prob(s,\varphi\wedge\psi) \big/ prob(s,\psi)$, the solution of multi until formula conditional probability primarily depends on the state meeting probability calculation of intersection operation path formula $\varphi_\cap = \varphi\wedge\psi$. For DTMRM M and given CPPRCTL multi until formula intersection operation $\varphi_\cap$, formula quantitative verification method of specified state s in DTMRM as follows:

First intersection operation formula $\varphi_\cap$ is represented as nondeterministic finite automaton $B_\cap$, 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. $P_s^M(\varphi_\cap) = P_{s^*}^{M\times B_\cap}(\Diamond_r^n accept)$. Where n and r respectively represent the transition step constraint two-dimensional array and transition resource consumption constraint two-dimensional array in $\varphi_\cap$ which all include two one-dimensional arrays.

V. Construction of Product Model

In this section, we give the formal definition of product model $M\times B_{\varphi_\cap}$ of nondeterministic finite automaton $B_{\varphi_\cap}$ corresponding to the multi until formula intersection operation and DTMRM M .

Definition 10 product model

Let $B_\cap = (\Sigma_\cap, Q_\cap, Q_{0\cap}, \delta_\cap, F_\cap)$ be the corresponding nondeterministic finite automaton of multi until formula intersection operation $\varphi_\cap$, $M = (S, P, AP, L, N, v)$ be a DTMRM, then their synchronization evolution product model denoted by $M\times B_{\varphi_\cap} = (S^*, P^*, AP^*, L^*, N^*, v^*)$. The symbols of $M\times B_{\varphi_\cap}$ are described below.

State set is expressed as:

$S^* = \{s^* = (s, q_\cap) | s\in S \wedge q_\cap \in Q_\cap\}$,

Transition probability is expressed as:

$$p^*((s,q_{\cap i}),n,(s',q_{\cap j})) = \begin{cases} p(s,n,s') & \text{if} \quad q_{\cap j}\in\delta_\cap(q_{\cap i}, L(s')) \\ 0 & \text{otherwise} \end{cases}$$

Labeling function is expressed as:

$$L^*(s,q_{\cap i}) = \begin{cases} L^*(s) \mathrm{U} \{accept\} & if\ q_{\cap i}\bigcap F_\cap \neq \varnothing \\ \varnothing & if\ \perp\in q_{\cap i} \\ L^*(s) & \text{otherwise} \end{cases}$$

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:

$$v^*(s,q) = \begin{cases} v(s) & if\ q\in\delta_\cap(q_{\cap 0}, L(s)) \\ 0 & \text{otherwise} \end{cases} .$$

After constructing, the meeting probability $P_s^M(\varphi_\cap)$ of multi until intersection formula $\varphi_\cap$ with transition constraints can be achieved by calculating accepting states reachable probability under the condition of transition step constraint and transition resource consumption constraint in the synchronous evolution product $M\times B_{\varphi_\cap}$.

The calculation process of accepting states reachable probability is described as follows:

(1)Use breadth-first algorithm to search state set of product model, and determine all the reachable paths that starting at initial state and end in acceptable state in the product model.

(2)Based on transition step constraints and transition resource consumption constraints in multi until formula, filter every reachable path and then determine constraint reachable path set.

(3)Find the same prefix path subset of constraint reachable path set by remove the invalid path set, then determine the final reachable path set and calculate path probability value.

Definition 11 longest same prefix path subset of path set

Let C be a finite path set, for each finite path σ in C , if it is the prefix path of some path which is not σ in C , then remove σ from C , and ultimately we can get the longest same prefix path subset of C .

VI. Instance Construction

In this section, based on computation method provided by this paper we give the detailed description of calculation process of multi until conditional probability formula $P(s_0, a\bigcup_{\le 30}^{\le 2} b | c\bigcup_{\le 40}^{\le 3} d)$ for the system model in Fig. 6.

According to the previous work in this paper, we can obtain:

$$P(s_0, a\bigcup_{\le 30}^{\le 2} b | c\bigcup_{\le 40}^{\le 3} d) = P(s_0, a\bigcup_{\le 30}^{\le 2} b \wedge c\bigcup_{\le 40}^{\le 3} d) \big/ P(s_0, c\bigcup_{\le 40}^{\le 3} d)$$

First, for multi until formula intersection operation $a\bigcup b\wedge c\bigcup d$ and multi until formula $c\bigcup d$, we construct nondeterministic finite automaton which can be shown in Fig. 5(3) and Fig. 5(2), then construct their synchronous evolution product model with original DTMRM model in Fig. 6. The product models are shown in Fig. 7 and Fig. 8, where initial state sets are respectively $\{s_0q_0q_0', s_0q_1q_0', s_0q_1q_1'\}$ and $\{s_0q_0', s_0q_1'\}$, accepting state sets are respectively $\{s_1q_2q_2', s_2q_2q_2', s_5q_2q_2'\}$ and $\{s_1q_2', s_2q_2', s_5q_2'\}$, and accepting state set are distinguished by dashed line ellipse in Fig. 7 and Fig. 8.

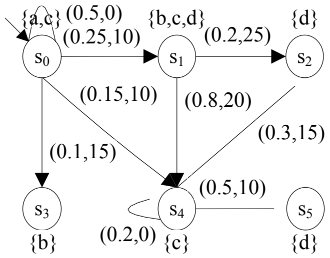


Figure 6. DTMRM model instance

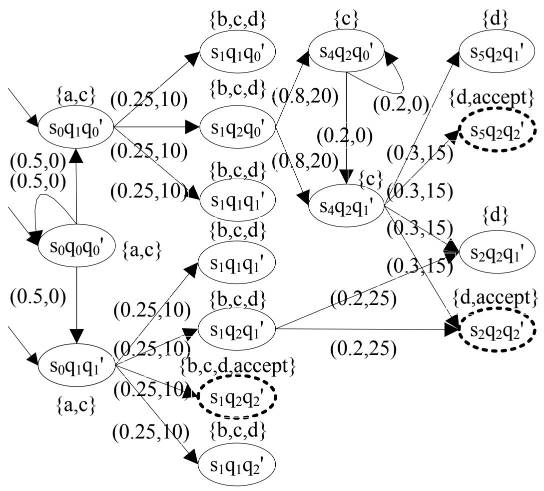


Figure 7. Product model of DTMRM and $a\bigcup b\wedge c\bigcup d$

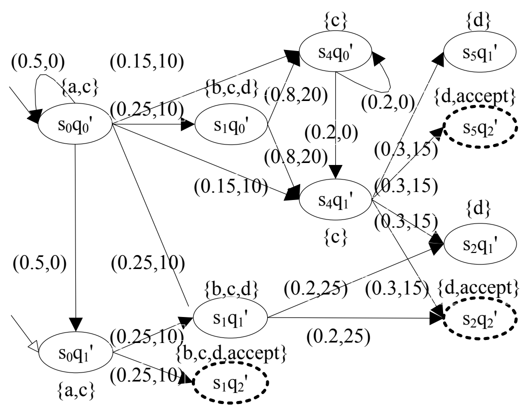


Figure 8. Product model of DTMRM and $c\bigcup d$

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

For product model of Fig. 7, the satisfying path set of path formula $a\bigcup b\wedge c\bigcup d$ starting at the initial state be expressed as:

$\{(s_0q_0q_0')^*(s_0q_1q_0')(s_1q_2q_2');$

$(s_0q_0q_0')^*(s_0q_1q_1')(s_1q_2q_1')(s_2q_2q_2');$

$(s_0q_0q_0')^*(s_0q_1q_0')(s_1q_2q_0')(s_4q_2q_0')^*(s_4q_2q_1')(s_5q_2q_2');$

$(s_0q_0q_0')^*(s_0q_1q_0')(s_1q_2q_0')(s_4q_2q_0')^*(s_4q_2q_1')(s_2q_2q_2')\}$

Similarly, for the product model in Fig. 8 , the satisfying path set of path formula $c\bigcup d$ starting at the initial state set of product model be expressed as:

$\{(s_0q_0')^*(s_0q_1')(s_1q_2');(s_0q_0')^*(s_0q_1')(s_1q_1')(s_2q_2');$

$(s_0q_0')^+(s_1q_0')(s_4q_0')^*(s_4q_1')(s_2q_2');$

$(s_0q_0')^+(s_1q_1')(s_2q_2');(s_0q_0')^+(s_1q_0')(s_4q_0')^*(s_4q_1')(s_5q_2');$

$(s_0q_0')^+(s_4q_0')^*(s_4q_1')(s_5q_2');(s_0q_0')^+(s_4q_0')^*(s_4q_1')(s_2q_2');\}$

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

Third step, we filter reachable path sets under the transition step constraints and transition resource consumption constraints in multi until formula $a\bigcup_{\le 30}^{\le 2} b\wedge c\bigcup_{\le 40}^{\le 3} d$ and $c\bigcup_{\le 40}^{\le 3} d$, and then obtain the longest same prefix path subset of constraint reachable path sets, that is the final satisfying path sets.

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

$\{(s_0q_1q_1')(s_1q_2q_1')(s_2q_2q_2');$

$(s_0q_0q_0')(s_0q_1q_1')(s_1q_2q_1')(s_2q_2q_2');$

$(s_0q_1q_1')(s_1q_2q_1')(s_2q_2q_2');$

$(s_0q_1q_0')(s_1q_2q_0')(s_4q_2q_1')(s_2q_2q_2')\}$

The final constraints satisfying path sets of $c\bigcup_{\le 40}^{\le 3} d$ are:

$\{(s_0q_0')(s_0q_1')(s_1q_2');(s_0q_0')(s_0q_0')(s_0q_1')(s_1q_2');$

$(s_0q_1')(s_1q_1')(s_2q_2');(s_0q_0')(s_0q_1')(s_1q_1')(s_2q_2');$

$(s_0q_0')(s_1q_0')(s_4q_1')(s_2q_2');(s_0q_0')(s_4q_0')(s_5q_2');$

$(s_0q_0')(s_0q_0')(s_4q_0')(s_5q_2');(s_0q_0')(s_4q_0')(s_2q_2');$

$(s_0q_0')(s_0q_0')(s_4q_0')(s_2q_2');(s_0q_0')(s_4q_0')(s_4q_0')(s_2q_2');$

$(s_0q_0')(s_4q_0')(s_4q_0')(s_5q_2');\}$

Last step, calculate the transition probability of final path sets and obtain the final calculation results: $P(s_0, a\bigcup_{\le 30}^{\le 2} b \,|\, c\bigcup_{\le 40}^{\le 3} d)$ =0.4204. The calculation results can be verified as correct by gradual analysis for DTMRM that be shown in Fig. 6.

VII. CONCLUSION AND FUTURE WORK

This paper aims at quantitative verification issues of conditional multi until formula property, draws on existing automaton property model checking method, introduces a specific property verification method of complex system conditional probability model, and the results of instance construction analysis show the effectiveness of the method. Next authors will further study the selection of constraint conditions interval and the details about algorithm optimization, and look for the broader applications of synchronization product model.

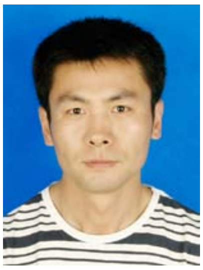
ACKNOWLEDGMENT

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] Yang Xu and Xiaoyao Xie, “Modeling and analysis of security protocols using colored petri nets,” *Journal of Computers*, vol. 6, no.1, pp.19-27, 2011.
[2] Huimin Lin and Wenhui Zhang, “Model checking: theories, techniques and applications,” *Acta Electronica Sinica*, vol. 30, no.l2, pp.1907-1912, 2002.
[3] Weibo Liu, Wenping Ma and Yuanyuan Yang, “Analysis and improvement of the ban modified andrew secure RPC protocol,” *Journal of Networks*, vol. 6, no. 4, pp.662-669, 2011.
[4] Huiling Shi, Wenke Ma, Meihong Yang and Xinchang Zhang, “A case study of model checking retail banking system with spin,” *Journal of Computers*, vol. 7, no.10, pp. 2503-2510, 2012.
[5] Wei Zhang, Wenke Ma, Huiling Shi and Fuqiang Zhu, “Model checking and verification of the internet payment system with spin,” *Journal of Software*, vol. 7, no. 9, pp. 1941-1949, 2012.
[6] Gueffaz M, Rampacek S and Nicolle C, “Temporal logic to query semantic graphs using the model checking method,” Journal *of Software*, vol. 7, no. 7, pp. 1462-1472, 2012.
[7] Jun Niu,Guosun Zeng and Wei Wang, “An approach of model checking time or space performance,” *Chinese Journal of Computers*, vol. 33, no. 9, pp.1621-1633, 2010.
[8] Conghua Zhou, Zhifeng Liu and Changda Wang, “Bounded model checking for probabilistic computation tree logic,” *Journal of Software*, vol. 23, no. 7, pp.1665-1668, 2012.
[9] Lijun Zhang, David N. Jansen, Flemming Nielson and Holger Hermanns, “Automaton-based CSL model checking,” *in Proceedings of 38th International Colloquium on Automaton, Languages and Programming*, Berlin:Springer, 2011, pp.271-282.
[10] Yang Gao, Ming Xu, Naijun Zhan and Lijun Zhang, “Model checking conditional CSL for continuous time Markov chains,” *Information Processing Letters*, 2013, pp. 44-50.
[11] Andrés M E and Van Rossum P, “Conditional probabilities over probabilistic and nondeterministic systems,” *Tools and Algorithms for the Construction and Analysis of Systems*, Springer Berlin Heidelberg, 2008, pp.157-172.
[12] Bing Guo, Yan Shen and Zili Shao, “The redefinition and some discussion of green computing,” *Chinese Journal of Computers*, vol. 32, no. 12, pp. 2311-2319, 2009.
[13] Xining Nie and Guoyong Cai, “Research on the rewriting logic model and its properties verification of petri net,” *Journal of Guilin University of Electronic Technology*, vol. 31, no. 3, pp. 208-212, 2011.
[14] Barate A, real-time interaction eith music structures in ieee 1599,” *Journal of Multimedia*, vol. 4, no. 1, pp.15-18, 2009.
[15] Shuiping Zhang and Huijune Luo, “The research of image encryption algorithm based on chaos cellular automaton,” *Journal of Multimedia*, vol. 7, no. 1, pp. 66-73, 2012.
[16] Lixing Li,Zhi Jin and Ge Li, “Modeling and verifying services of internet of things based on timed automaton,” *Chinese Journal of Computers*, vol. 34, no. 8, pp.1366-1377, 2011.
[17] Cong Wang and Zhixue Wang, “Automaton theory based model checking method for uml activity diagram,” *Journal of System Simulation*, vol. 19, no. 22, pp.5311-5314, 2007.
[18] Lihui Lei and Zhenhua Duan, “An extended deterministic finite automaton based method for the verification of composite web services,” *Journal of Software*, vol. 18, no. 12, pp.2980-2990, 2007.

Mingyu Ji 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 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.