Temporal Logic of Stochastic Actions for Verification of Probabilistic Systems

LI Jun-tao
Information College
Guizhou University of Finance and Economic
Guiyang, china
E-mail: jtxq@qq.com

LONG Shi-gong,
Computer Science and Technology College
Guizhou University
Guiyang, china
E-mail: 526796467@qq.com

Abstract-The specification and verification of probabilistic systems were usually based on Computational Tree Logic, and systems and properties were specified by different language respectively. This paper extends and reforms Temporal Logic of Actions, puts foreword Temporal Logic of Stochastic Actions (TLSA), which can use additional state-action probabilistic distribution and probabilistic operator to specify probabilistic systems and their properties in the same logic.

Keywords-Specifying systems; System verification; TLA; Probabilistic systems

I. Introduction

As soon as the Model Checking was invented in 1980's, researchers had started applying it to the study of Probabilistic Systems' verification. In the beginning, people focus on the qualitative properties of system, for example, the program is whether or not terminating with probability 1. Afterwards, the algorithms for verifying quantitative properties have been progressed also. At present, the verification technology of Probabilistic Systems is mainly used for the field of security, distributed algorithms, systems biology, and system performance analysis, and so on.

In past thirty years, the implementation model of Probabilistic Systems is mainly based on Markov decision processes [1][2](MDPs), there are also some others import timed automata[6], putdown automata[7], or two-player game. They specify the property of system with linear temporal logic (LTL), ω-regular properties or probabilistic computation tree logic (PCTL), the latter imports probability distribution to computation tree logic (CTL); it is most used description language of Probabilistic Systems. Obviously, the traditional research method used different description language to specify the models or properties of Probabilistic Systems; this is not well for the property verification and design implementation of system.

This paper puts forward Temporal Logic of Stochastic Actions (TLSA), it is extension of Temporal Logic of Actions [4] (TLA) with probability, the latter is based on linear temporal logic, it defined the actions and operators, and can achieve specifying and verification of concurrent systems and their properties; the former inherit the most prominent feature of TLA: system and its properties can be described using TLSA at the same time.

II. Probabilistic Transition system

Probabilistic Transition Systems (PTSs) is abstract model of Probabilistic Systems, its definition is followed:

Defination1.1 Probabilistic Transition systems is a 5-tuple: $\mathcal{P} = \{\mathcal{S}, \mathcal{I}, \mathcal{A}, \delta, \lambda\}$,among them:

$\mathcal{S}$：States set of system;

$\mathcal{I}$：Initial states set of system;

$\mathcal{A}$：Actions set of system;

δ ： $\mathcal{S}\times\mathcal{A}\rightarrow\mathcal{S}$ relationship of states trasation;

λ：（$\mathcal{S}\times\mathcal{A}\rightarrow\mathcal{S}$）$\rightarrow[0,1]$ is probability distribution of system actions, and meet the condition: $\forall s \in \mathcal{S}$, $\sum_{A\in\mathcal{A}} p(s,A,s') = 1$.

We can see that probabilistic transition systems are Label Transition Systems (LTS) adding a probability distribution of actions.

III. Syntax and Semantics ofTLSA

3.1 Syntax

TLSA's symbol includes:

（1） Probabilistic values: *pr*(∈[0,1]);
（2） Constant symbol: c, c_1, c_2 ...
（3） Rigid variables: u, u_1, u_2 ...
（4） Flexible variable: x, x_1, x_2 ...
（5） Atomic proposition: p, p_1, p_2 ...
（6） Constant element symbol: m, m_1, m_2 ...
（7） Arithmetic operator: +,-, *;
（8） Relational operators: <, =;
（9） logical operator: ∧ , ¬ , $><pr;$
（10） Quantifier: ∃ ;
（11） Temporal operator：′, [].

The other conjunctions, for example ∨,

978-1-4673-6593-2/15 $31.00 © 2015 IEEE
DOI 10.1109/DCABES.2015.23

⇒, ⇔, and so on, can be defined by $\wedge$ and $\neg$; Temporal operator $\Diamond$ can be defined by $\Box$ and $\neg$.

Defination2.1 *State:* Once assignment for all variables in system. All those possible assignments constitute the states set of system, named St . Initial states of system *Φ* write as $Init_{\Phi}$.

Defination2.2 *State functions* [4]:

$$f \triangleq c \mid u \mid x \mid f_1 + f_2 \mid f_1 - f_2 \mid f_1 * f_2$$

c is a constant, *u* is a rigid variable, *x* is a flexible variable.

Defination2.3 *State predicates* [4]:

$$q \triangleq p \mid (f_1 = f_2) \mid (f_1 < f_2) \mid \neg q \mid (q_1 \wedge q_2) \mid \exists uq$$

P is a atomic proposition, f_1 and f_2 are states functions, *u* is a rigid variable.

Defination2.4 *Action functions*:

$$f \triangleq c \mid u \mid x \mid x' \mid (f_1 + f_2) \mid (f_1 - f_2) \mid (f_1 * f_2)$$

c is a constant, *u* is a rigid variable, *x* is a flexible variable, *x*' is the value of *x* in new state, ， f_1 and f_2 are functions of actions.

Defination2.5 *Actions*[4]:

$$A \triangleq p \mid p' \mid (f_1 = f_2) \mid (f_1 < f_2) \mid \neg A \mid (A_1 \wedge A_2) \mid \exists uA$$

p is a atomic proposition, *p′* is the value of *p* in new state，f_1 and f_2 are functions of actions, *u* is a rigid variable.

PS: All actions constitute the action set of system，named 。

Defination2.6 *State actions probability distribution:* if

$$p(s, A) \triangleq \begin{cases} pr, A \in \mathcal{A} \text{ 且 } s \in St \\ 0, \text{ 其它} \end{cases}$$

for $\forall s \in \mathcal{S}$ meet

$$\sum_{A \in \mathcal{A}} p(s, A) = 1$$

then $p(s, A)$ is state actions probability distribution. We write state actions probability distribution of present state as *p*(·), and the probability of action *A* at present state as *p*(·, *A*).

Defination2.7 *Probabilityoperator* $_{\geq\leq pr}$: is a probability range of a predicate, action, or another Boolean expression, $\geq\leq$ is one of <, =, ≥, ≤, >, $pr \in [0,1]$ is probability value. For example, $_{=0.4}(A)$ represents action *A* occur by probability of 0.4.

Defination2.8 *Probabilistic Actions*:

$$M \overset{p(\cdot)}{=} A_1 \vee A_2 \vee \cdots \vee A_n$$

p(·) is actions probability distribution of present state, A_i（$i \in$ ）is action,

Defination2.9 *Stuttering*:

$$[A]_v \triangleq A \vee (v' = v) \qquad （2.1）$$

$$\langle A \rangle_v \triangleq A \wedge (v' \neq v) \qquad （2.2）$$

A is a action, *v* is a tuple constituted by state variable, *v′* is value of *v* in next state.

$$[M]_v \triangleq M \vee (v' = v) \qquad （2.3）$$

$$\langle M \rangle_v \triangleq M \wedge (v' \neq v) \qquad （2.4）$$

Defination2.10 *Simple TSLA formula*:

$$F \triangleq P \mid \Box P \mid \Box[A]_v \mid \neg F \mid (F_1 \wedge F_2)$$

P is state predicate, *A* is a action.

Defination2.11 *Enabled*:

If an action A, its probability satisfy $p(\cdot, A) > 0$, then *A* is enable at present state., marked *Enabled<A>*$_v$. *Enabled* is called Enable Predicate.

Defination2.12 *Fairness*:

$$WF_v(A) \triangleq \Box\Diamond\neg ENABLED\langle A \rangle_v \vee \Box\Diamond\langle A \rangle_v$$

$$SF_v(A) \triangleq \Diamond\Box ENABLED\langle A \rangle_v \vee \Box\Diamond\langle A \rangle_v$$

WF$_v$(*A*) and *SF*$_v$(*A*) are called Fairness together, write as *F*$_v$(*A*).

Defination2.13 TLSA formula:

$$\Phi \triangleq Init_{\Phi} \wedge \Box[M]_v \wedge F_v(A_1) \wedge \cdots \wedge F_v(A_n)$$

$Init_{\Phi}$ are initial states of system, *M* is probability action, A_i（$i \in \mathbb{N}$) is action.

3.2 SEMANTICS

we use symbol “$\llbracket\ \rrbracket$” represent semantics, for example, $\llbracket A \rrbracket$ represent the semantics of action *A*.

According to the definition of actions, it can include the value of variable in nest state, it means that action is a relationship between two state:

$$s\llbracket A \rrbracket t \triangleq A(\forall v : s\llbracket v \rrbracket / v, t\llbracket v \rrbracket / v')$$

State predicate *P* can regard as action that don’t include new value of variable, this means its semantics is irrelevant with next state. Wecan write:

$$s\llbracket P \rrbracket \triangleq P(\forall v : s\llbracket v \rrbracket / v)$$

Probability action *M* is disjunction operation with a series of actions under the conditions of probability distribution, its result still is an action:

$$s\llbracket M \rrbracket t \triangleq M(\forall v : s\llbracket v \rrbracket / v, t\llbracket v \rrbracket / v')$$

Defination2.14 *Behaviors* [4]: Behavior is a sequence constituted by unlimited state, marked σ , ith state write as σ_i :

$$\sigma \triangleq \sigma_0 \xrightarrow{M_0} \sigma_1 \xrightarrow{M_1} \sigma_2 \xrightarrow{M_2} \cdots$$

In followed text, we will use $\sigma[..n]$ and σ^{+n} presenting top n state limited sequence or followed unlimited sequence after nth state respectively:

$$\sigma[..n] \triangleq \sigma_0 \xrightarrow{M_0} \sigma_1 \xrightarrow{M_1} \cdots \xrightarrow{M_{n-1}} \sigma_n$$

$$\sigma^{+n} \triangleq \sigma_n \xrightarrow{M_n} \sigma_{n+1} \xrightarrow{M_{n+1}} \sigma_{n+2} \xrightarrow{M_{n+2}} \cdots$$

Of course, σ can be write as $\sigma[..n] \circ \sigma^{+n}$ 。

To state predicate ***P,*** we have：

$$\sigma[\![P]\!] \triangleq \sigma_0[\![P]\!]$$

$$\sigma^{+n}[\![P]\!] \triangleq \sigma_n[\![P]\!]$$

$$\sigma[\![\Box P]\!] \triangleq \forall n \in N : \sigma^{+n}[\![P]\!]$$

To action ***A***:

$$\sigma[\![A]\!] \triangleq \sigma_0[\![A]\!]\sigma_1$$

$$\sigma^{+n}[\![A]\!] \triangleq \sigma_n[\![A]\!]\sigma_{n+1}$$

$$\sigma[\![\Box A]\!] \triangleq \forall n \in N : \sigma^{+n}[\![A]\!]$$

To probability action ***M***:

$$\sigma[\![M]\!] \triangleq \sigma_0[\![M]\!]\sigma_1$$

$$\sigma^{+n}[\![M]\!] \triangleq \sigma_n[\![M]\!]\sigma_{n+1}$$

$$\sigma[\![\Box M]\!] \triangleq \forall n \in N : \sigma^{+n}[\![M]\!]$$

Similarly, those are the semantics of other concept below:

Eventually: $[\![\Diamond F]\!]$ means formula F can be true eventually. In other words, at present or later, there is a state *s*: $s \models F$. *In fact ,* $\Diamond F \triangleq \neg\Box\neg F$. Hence it's not hard to get:

$$\sigma[\![\Diamond F]\!] = \exists n \in N : \sigma^{+n}[\![F]\!]$$

Infinitely often: If formula F is true at unlimited states in behavior σ , then we can say $\sigma \models \Box\Diamond F$. In the same, the formal semantics of $\Box\Diamond F$ can be defined below:

$$\sigma[\![\Box\Diamond F]\!] \triangleq \forall n : (\exists m \in N : \sigma^{+n+m}[\![F]\!])$$

Leads to: formula $\Box(F \to \Diamond G)$ is true, if and only if that F is true must lead to G is true at later state. Its formal semantics is:

$$\sigma[\![\Box(F \to \Diamond G)]\!] \triangleq \forall n : (\sigma^{+n}[\![F]\!] \Rightarrow (\exists m \in N : \sigma^{+(n+m)}[\![G]\!]))$$

3.3 Probabilistic property

Besides common qualitative properties of concurrent system, such as *invariance*, *eventuality* [4], we also analysis the quantitative properties of probabilistic system.

The probabilistic system defined in defination1.1 is a label transition system with probability distribution of action, so we can specify and verify some probabilistic properties of system actions. For example, after action A occurring, the probability of action B occur eventually is not less than 0.4, that is $\mathbb{P}_{\geq 0.4}(A \to \Diamond B)$. Those properties are mainly used to describe the reliability and performance of systems.

IV. SPESIFYING PROBABILISTIC SYSTEM BY TLSA

We observe a biological group, use x representing the quantity of that group, supposing the biggest quantity is 5, at next moment the quantity may be three kind variation below: increasing 1, reducing 1, not change; and besides extreme states of x=0 and x=5, the probability of three kind variation is 1/3; at state of x=0, the probability of no change is 1; at the state of x=5, the probability of reducing 1 is 1. Its probabilistic transition graph is as figure 1.

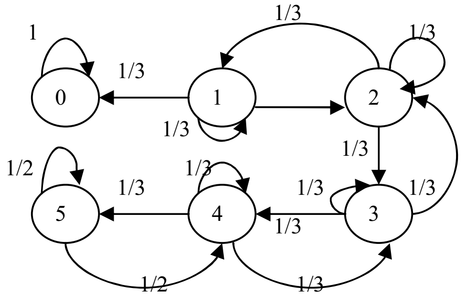

Figure 1. Biological group quantity transition graph

Figure 1 indicate a classical probabilistic system, its state space $\mathcal{S} \triangleq \{x \mid x = 0,1,2,3,4,5\}$, $Init_\Phi \triangleq x = 3$, three actions are: $A \triangleq x' = x+1$, $B \triangleq x' = x-1$ and $C \triangleq x' = x$; probability distribution of actions is: at state 1, 2, 3, 4, the probability of three actions occurring is equal to 1/3; at state 0, the probability of action C is 1; at state 5, the probability of action B and C are all 1/2.

We specify the probabilistic system as figure 1 using TLSA below:

----------------MODULE GenExt------------------

EXTENDS Naturals, Reals

VARIABLE x

$GEini \triangleq x = 3$

$GEnxtA \triangleq x' = x+1$

$GEnxtB \triangleq x' = x-1$

$GEnxtC \triangleq x' = x$

IF $s \neq 0, s \neq 5, a = A, B, C$ THEN $p(s,a) \triangleq 1/3$

IF $s = 5, a = B, C$ THEN $p(s,a) \triangleq 1/2$

IF $s = 0, a = C$ THEN $p(s,a) \triangleq 1$

$GEnxt \overset{p(\cdot)}{=} \vee GEnxtA$

$\qquad \vee GEnxtB$

$\qquad \vee GEnxtC$

$GE \triangleq GEini \wedge \Box[GEnxt]_x \wedge WF_x(GEnxtA)$

V. MODEL CHECKING OF PROBABILISTIC PROPERTIES

If the system as figure 1 satisfy that the probability of action A occurring lead to action B occurring is 0.5, that is $\mathbb{P}_{\leq 0.5}(A \to <> B)$.

We can embed probability module in system specification by TLSA.

```
-----------------MODULE
GenExt------------------
EXTENDS Naturals, Reals
VARIABLE $x$
CONSTANT $Rpr$
ASUME $Rpr \in [0,1]$
  -----------------MODULE Prob-----------------
  VARABLE $pr$
   $Pini \triangleq pr = 0$
   $Pnxt \triangleq pr' = IF\ GEnxtB\ THEN\ \ p(\cdot, GEnxtB)$
            $ELSE\ IF\ \ pr = 0\ \ THEN\ 1 - p(\cdot, GEnxtB)$
             $ELSE\ \ pr * (1 - p(\cdot, GEnxtB))$

   $GEProb \triangleq Pini \wedge \Box[Pnxt]_{pr}$
  ------------------------------------------------------
$P(pr) \triangleq INSTANCE\ Prob$
$GEini \triangleq x = 3$
$GEnxtA \triangleq x^{'} = x + 1$
$GEnxtB \triangleq x^{'} = x - 1$
$GEnxtC \triangleq x^{'} = x$
IF $s \neq 0, s \neq 5, a = A, B, C$  THEN $p(s,a) \triangleq 1/3$
IF $s = 5, a = B, C$ THEN $p(s,a) \triangleq 1/2$
IF $s = 0, a = C$ THEN $p(s,a) \triangleq 1$
$GEnxt \overset{p(\cdot)}{=} \vee GEnxtA$
             $\vee GEnxtB$
             $\vee GEnxtC$
$GE \triangleq GEini \wedge \Box[GEnxt \wedge P(pr)!Pnxt]_x \wedge WF_x(GEnxtA)$

---------------------------------------------------------
---
THEOREM  $GE \Rightarrow \mathbb{P}_{\leq 0.5}(GEnxtA \rightarrow \Diamond GEnxtB)$
```

The model checking tools of TLSA can be obtained by adding probabilistic logic operator and definition of probability calculation module, we don't discuss it here.

VI. Conclusion

Based on Temporal Logic of Actions, we put forward a Temporal Logic of Stochastic Actions (TLSA), by adding state action probability attribution and probabilistic operator; it is fit for specifying and model checking probabilistic systems. Compared with Probabilistic Computation Tree Logic (PTCL)[2], Continuous Stochastic Logic (CSL)[3] and Real Timed Probabilistic Computation Tree Logic(RPTCL)[7], the most important feature of TLSA is that it can describe system , and it can describe the properties of system in same time. This is useful for verification of system properties and refinement of system design.

In addition, the probability distribution is based on system actions in TLSA, and the quantity of actions are fewer than system state, this is beneficial to specifying and calculating of probabilistic properties.

The model checking tools of TLSA can be obtained by extending TLC.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NO. 61163001/F020106), Natural Science Foundation of Guizhou Province (NO.J [2012]2096), and High-level Talent Recruitment Foundation of Guizhou University of finance and Economics.

References

[1] Katoen, J.P. Perspectives in probabilistic verification. Proceedings 2nd IEEE/IFIP International Symposium on Theoretical Aspects of Software Engineering, Nanjing, China, 2008.

[2] C. Baier and J.-P. Katoen. Principles of Model Checking. MIT Press, 1st edition, 2008.

[3] Baier, Christel and Cloth, Lucia and Haverkort, Boudewijn R. and Kuntz, Matthias and Siegle, Markus. Model Checking Markov Chains with Actions and State Labels. IEEE Transactions on Software Engineering, 33 (4), 2007. pp. 209-224.

[4] Leslie Lamport. The temporal logic of actions. ACM Transactions on Programming Languages and Systems (TOPLAS), v.16 n.3, p.872-923, 1994.

[5] Luca De Alfaro , Zohar Manna, Formal verification of probabilistic systems, Stanford University, Stanford, CA, 1998.

[6] A. Kucera, J. Esparza, R. Mayr. Model checking probabilistic pushdown automata. LMCS 2(1): (2006).

[7] M.Z. Kwiatkowska, G. Norman, R. Segala, J. Sproston: Automatic verification of real-time systems with discrete probability distributions. Theor. Comput. Sci. 282(1): 101-150, 2002.