On Actions of Temporal Logic of Actions

Juntao Li, Ziyi You, Zhengyi Tang, Xiang Li
College of Computer & Information
Guizhou University
Guiyang, China
e-mail: jtxq@21cn.com

***Abstract*—Introducing actions into LTL is most outstanding characteristic of Temporal Logic of Actions. We argue actions' properties, and study fairness, liveness and safety of system via the action's properties in Temporal Logic of Actions, put forward safety and liveness of action, redefine the liveness and safety of concurrent systems from actions' view, give a hint to system verification.**

Keywords-action; liveness; syfety; fairness; TLA

I. INTRODUCTION

As hardware and software systems become more and more complex, the more sophisticated methods of verification are required. Temporal logic of actions (TLA) is one of them. It is based on linear temporal logic (LTL), and applicable to concurrent systems' specifying and verification. Its most outstanding characteristic is that systems and their properties can both be specified in TLA formulas.

The main contributions of TLA due to two points: First, it introduces "actions" into LTL, witch can connect states from previous to next; Second is "stuttering steps", it permits all variables used describing systems changeless from one state to next. Its significance is the same to zero for nature numbers. In fact, stuttering steps are special actions. In this paper, we will study actions in-depth, and reargue properties of concurrent systems from actions view, redefine liveness and safety of concurrent system, and prove the new definitions equivalent to the old ones.

II. TEMPORAL LOGIC OF ACTIONS

We now present a simple and selective introduction of TLA, for more details expression, refer to [1] and [5].

TLA is a first-order temporal logic. It's a combination of linear temporal logic and logic of actions. TLA inherits unary modal operator □ that reads *always* from temporal logic. A formula in TLA is constructed by using Boolean operators and the modal operator by following the familiar syntactic rules. The dual of □ operator is ◇ and defined as follows: ◇F ≡ ¬□¬F. Therefore, □F means "F is always (i.e. at all times in the future) the case" whereas ◇F means "F is eventually (i.e. at some time in the future) the case".

The semantic of TLA is based on *states.* We will use *behaviors* to give the semantic meaning of $[\![\Box F]\!]$ for a formula F in TLA.

States: A state is an assignment of values to variables used to describe system.

Behaviors: A *behavior* σ is an infinite sequence of states, σ_i stand for (i+1)st state of the sequence:

$$\sigma \triangleq \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \sigma_2 \xrightarrow{A_2} \cdots$$

Let's define $\sigma[..n]$ and σ^{+n} to be the prefix and suffix of σ respectively:

$$\sigma[..n] \triangleq \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$$

$$\sigma^{+n} \triangleq \sigma_n \xrightarrow{A_n} \sigma_{n+1} \xrightarrow{A_{n+1}} \sigma_{n+2} \xrightarrow{A_{n+2}} \cdots$$

Accordingly, $\sigma = \sigma[..n] \circ \sigma^{+n}$.

Semantic meaning of $[\![\Box F]\!]$:

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

Furthermore, we can express several usual characteristics in this language, too. Given the semantics, we can represent some properties by using this language. Let us reproduce several of them here in order to make the intuition behind the semantics clearer.

Eventually: The formula $[\![\Diamond F]\!]$ asserts that F is eventually true. Considering the previously given definition of ◇, we can verify the following definition.

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

Infinitely Often: If F is true at infinitely many times during the behavior, then the behavior satisfies $\Box\Diamond F$. By the previous definitions, it can be seen that the definition of $\Box\Diamond F$ is as follows.

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

Leads To: The formula is true if and only if at any time when F is true, G is true at that time or at some later time. We leave it to the reader to convince herself that this formula in fact represents the given statement.

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

Some fundamental conceptions of TLA as follows:

***State predicates*:** A state predicate is a Boolean expression constructed from variables and constant symbols. The validity of the predicate P at state s will be denoted by $s \models P$ or $s[\![P]\!]$.

***Primed variables*:** A primed variable v' is variable v evaluating the next state.

Actions: An action is a Boolean-valued expression containing constant symbols, variables, and primed variables. In fact, an action is a special predicate which assert the relation between old state and new one.

Semantic meaning of action A is:

978-1-4244-6542-2/$26.00 © 2010 IEEE

$s[\![A]\!]t \triangleq A(\forall v : s[\![v]\!]/v, t[\![v]\!]/v')$

We can regard state predicates as actions without primed variable. To a state predicate P, $s[\![P]\!]t \equiv s[\![P]\!]$.

A step **:** <*s, t*> is a *A step* iff

$[\![A]\!] = A(\forall v : s[\![v]\!]/v, t[\![v]\!]/v')$ is true. If <*s, t*> is a *A* step, then we represent it by $s[\![A]\!]t$.

Stuttering step: If system transits from state *s* to *t,* all variables interested are unchanged, then we call <*s, t*> a stuttering step. In fact, t and s are same state of system we specify.

Enabled predicates: An action *A* is *enabled* for a state *s,* iff there is a state *t,* and $s[\![A]\!]t$ holds. If *A* is enabled for *s,* then $s[\![Enabled\ A]\!]$. We call (Enabled *A*) *Enabled predicate*. It is also written $ENABLED\langle A\rangle_v$ in TLA.

Being similar to the semantic meaning of $[\![\Box F]\!]$, we can present the semantic meaning of actions and state predicates. To a state predicate *P* and a action *A:*

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

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

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

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

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

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

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

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

A TLA ***formula*** is a predicate of a behavior. Elementary TLA formulas should include P、$\Box P$、$\Diamond P$、A、$\Box A$、$\Diamond A$ and $Enabled\ A$. Here P is a state predicate, A a action, and $Enabled\ A$ an enabled predicate. However, as we have known, the system specified by TLA permit the stuttering steps. In order to denoting the stuttering steps, TLA introduces two notations of $\langle A\rangle_v$ and $[A]_v$:

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

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

Here v is a state function. In TLA, a real A step is denoted by $\langle A\rangle_v$, and $[A]_v$ denotes a real A step or a stuttering step.

To A in enabled predicate of $Enabled\ A$, being replaced by $[A]_v$ is meaningless. So we rewrite $Enabled\ A$ as $Enabled\langle A\rangle_v$.

III. SOME TEMPORAL TAUTOLOGIES

For the sake of the completeness of this paper, we briefly mention some temporal tautologies of TLA. They are given as follows:

$\Box F \Rightarrow F$

$F \Rightarrow \Diamond F$

$\neg\Box F \Rightarrow \Diamond\neg F$

$\Box(F \wedge G) \equiv (\Box F) \wedge (\Box G)$

$\Diamond(F \vee G) \equiv (\Diamond F) \vee (\Diamond G)$

$(\Box F) \vee (\Box G) \Rightarrow \Box(F \vee G)$

$\Diamond(F \wedge G) \Rightarrow (\Diamond F) \wedge (\Diamond G)$

$\Box\Diamond(F \vee G) \equiv (\Box\Diamond F) \vee (\Box\Diamond G)$

$\Diamond\Box(F \wedge G) \equiv (\Diamond\Box F) \wedge (\Diamond\Box G)$

$\Box(F \Rightarrow G) \equiv (\Box F) \Rightarrow (\Box G)$

$\Diamond\Box F \Rightarrow \Box\Diamond F$

We select j) to prove, the proof of the others can be found in cited literature [1].

Proof:

$\Diamond\Box F \equiv \exists m \in N : (\forall n \in N : \sigma^{+m+n} \models F)$

$\Rightarrow \forall m \in N : (\exists n \in N : \sigma^{+m+n} \models F)$

$\equiv \Box\Diamond F$

The $\Rightarrow$ is because $\exists m \in N : \forall n \in N : A(n,m)$

$\Rightarrow \forall m \in N : \exists n \in N : A(n,m)$

Q.E.D

IV. PROPERTIES OF ACTIONS

As we know, introducing actions into LTL is most outstanding characteristic of TLA. But in study of properties of system specified by TLA, pioneers have not made use of them. In this section, we argue the properties of actions, and reargue the properties of systems using them in later section.

The aim that we study the properties of actions is recheck the properties of system. Before arguing the property of actions, we present the definitions of behaviors' properties.

Properties of behavior: A property Φ is a set of behaviors, i.e.

$$\Phi \triangleq \left\{\sigma \mid \sigma = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots; \sigma_i \in St, A_i \in \mathcal{A}\right\}$$

Here St is states set containing all states of system and $\mathcal{A}$ a actions set including all actions of system we describe. We write $\sigma \in \Phi$ or $\sigma \models \Phi$ to present the fact behavior σ in set of behaviors of Φ.

To finite sequences as:

$\sigma[..n] = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$, $\sigma[..n] \models \Phi$ if and only if existing infinite sequence:

$\sigma^{+n} \triangleq \sigma_n \xrightarrow{A_n} \sigma_{n+1} \xrightarrow{A_{n+1}} \cdots$ and

$\sigma[..n] \circ \sigma^{+n} \in \Phi$.

We present properties of actions as follows.

A. Liveness of Actions

A action A is live in a state s, if and only if A is enabled at s, i.e. $s[\![Enabled\ A]\!]$.

Liveness of actions is only a possibility of actions occurring, that whether they can really occur, we will describe by fairness.

B. Fairness of Actions

Weak fairness (justice): A behavior $\sigma = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots$ is weak fairness w.r.t an action $A \in \mathcal{A}$, iff the following condition holds:

If A is enabled at all states beyond m then $A_n = A$ for some $n \geq m$.

We can write it as TLA formula:

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

Strong fairness (compassion): A behavior $\sigma = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots$ is strong fairness w.r.t an action A, iff the following condition holds:

If A is enabled at infinitely many states beyond m then $A_n = A$ for some $n \geq m$.

We write it as TLA formula:

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

C. Safty of Actionss

An action A is safety for a property Φ, if and only if the following condition holds:

For an arbitrary finite sequence :

$\sigma[..n] = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n \models \Phi$, if A is enabled at σ_n, i.e. $\sigma_n [\![Enabled\ A]\!]$, and

$\sigma[..(n+1)] = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n \xrightarrow{A} \sigma_{n+1}$

then $\sigma[..(n+1)] \models \Phi$. Here we call A is ***safe action*** of Φ.

D. Sustainability

An action A is sustainable in a behavior σ, if and only if A is enabled until A occurs in σ.

We write it as TLA formula:

$Enabled\langle A\rangle_v \Rightarrow \Box Enabled\langle A\rangle_v \vee \langle A\rangle_v$

E. Reversibility

An action A is reversible in a behavior σ, if and only if following condition holds:

At an arbitrary state $s \in \sigma$, if A is enabled and A occurs, at next state t exists another action B is enabled, and B occurring would transfer t back to s.

The semantic meaning of reversibility is:

$(s \models Enabled\langle A\rangle_v) \wedge s[\![A]\!]t \Rightarrow$

$\exists B : ((t \models Enabled\langle B\rangle_v \wedge t[\![B]\!]s)$

V. Relationship Beteen Actions

A. Causality

A and B are two actions. At an arbitrary state $s \in \sigma$, A is enabled and B disenabled, if A occurs, and at next state t, B is enabled, then we say B is caused by A in σ.

We write it as TLA formula:

$Enabled\langle A\rangle_v \wedge \neg Enabled\langle B\rangle_v \wedge \langle A\rangle_v \Rightarrow Enabled\langle B\rangle_v$

B. Concurrent

If two actions A and B are both enabled at an arbitrary state $s \in \sigma$, and A occurs, at new state B is still enabled and vice versa, then A and B are concurrent in σ. This can be written as TLA formula:

$Enabled\langle A\rangle_v \wedge Enabled\langle B\rangle_v \wedge \langle A\rangle_v \Rightarrow Enabled\langle B\rangle_v$, and

$Enabled\langle A\rangle_v \wedge Enabled\langle B\rangle_v \wedge \langle B\rangle_v \Rightarrow Enabled\langle A\rangle_v$

C. Competition

Two actions A and B are both enabled at an arbitrary state $s \in \sigma$, if A occurs, then B is disenabled at next state t, and vice versa, we say that A and B are competitive in σ.

We specify this relationship in TLA formulas as follows:

$Enabled\langle A\rangle_v \wedge Enabled\langle B\rangle_v \wedge \langle A\rangle_v \Rightarrow \neg Enabled\langle B\rangle_v$, and

$Enabled\langle A\rangle_v \wedge Enabled\langle B\rangle_v \wedge \langle B\rangle_v \Rightarrow \neg Enabled\langle A\rangle_v$

D. Inversion

An action A is enabled at an arbitrary state $s \in \sigma$, if $s[\![A]\!]t$, and exists an action B enabled at t, and $t[\![B]\!]s$, or vice versa, then A and B are mutually inverse. Here we can write B as A^{-1} or A as B^{-1}.

VI. Actions and Behaviors

Verifying properties of concurrent system contain two classes: liveness and safety, and liveness is usually expressed by fairness. Properties of liveness and safety refer to behaviors' properties. They are different from ones of actions. In this section we will ague them based on actions. In order to distinguish between the two kind of properties, we call former behaviors' properties.

First, we present the traditional definition of liveness and safety of behaviors:

Definition 1. (Alpern, Schneider)

A property Φ is a safety property iff the following condition holds:

$\sigma = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \sigma_2 \xrightarrow{A_2} \cdots$ is in Φ if and only if every finite prefix:

$\sigma[..n] = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$

can be extended infinite sequence:

$\sigma_0 \xrightarrow{A_0} \cdots \xrightarrow{A_{n-1}} \sigma_n \xrightarrow{B_n} \sigma_{n+1} \xrightarrow{B_{n+1}} \cdots \in \Phi$

A property Φ is a liveness property iff any finite sequence $\sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$ can be extended infinite sequence:

$\sigma_0 \xrightarrow{A_0} \cdots \xrightarrow{A_{n-1}} \sigma_n \xrightarrow{B_n} \sigma_{n+1} \xrightarrow{B_{n+1}} \cdots \in \Phi$.

Definition 1 has an attractive simplicity, but it not easy to understand sometimes. We present a new definition of liveness an safety using actions:

Definition 2. A property Φ is a **safety property** iff the following condition holds:

$\sigma = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \sigma_2 \xrightarrow{A_2} \cdots$ is in Φ if and only if every finite prefix:

$\sigma[..n] = \sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$

exists safe action of Φ at is enabled at σ_n.

A property Φ is a **liveness property** iff any finite sequence $\sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$, exists safe action of Φ is enabled at σ_n.

According the definition of safe action, Definition 2 and Definition 1 is equivalent obviously, and definition 2 is more intuitionistic than definition 1. We prove the equivalence of them as follows. Here we prove only two liveness properties are equivalent, the proof of two safeties are similar.

Proof: By definition 1, if any finite sequence $\sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$ can be extended infinite sequence:

$\sigma_0 \xrightarrow{A_0} \cdots \xrightarrow{A_{n-1}} \sigma_n \xrightarrow{B_n} \sigma_{n+1} \xrightarrow{B_{n+1}} \cdots \in \Phi$

According the definition of safe action, B_n is a safe action for Φ, i.e. existing safe action of Φ is enabled at σ_n. Contrarily, if existing safe action of Φ is enabled at σ_n w.r.t. any finite sequence $\sigma_0 \xrightarrow{A_0} \sigma_1 \xrightarrow{A_1} \cdots \xrightarrow{A_{n-1}} \sigma_n$, then it can be can be extended infinite sequence:

$\sigma_0 \xrightarrow{A_0} \cdots \xrightarrow{A_{n-1}} \sigma_n \xrightarrow{B_n} \sigma_{n+1} \xrightarrow{B_{n+1}} \cdots \in \Phi$ by the meaning of safe action. Q.E.D

In addition, TLA attaches importance to fairness, when we specify system using the specifying language TLA+, almost all module does make use of fairness conditions. Nevertheless, those fairness refer to actions' fairness. TLA doesn't distinguish actions' fairness and behaviors' fairness. We propose definition of behaviors' fairness as follows.

Definition 3. A behavior σ is a ***weakly fair behavior***, if it is at least weakly fair w.r.t any action.

A behavior σ is a ***strong fair behavior***, if it is strongly fair w.r.t any action.

VII. CONCLUSION

The notion of actions is significant for TLA, but it hasn't received recognition from former pursuers. We study actions in-depth, analysis properties of them, distinguish them from properties of behaviors, and redefine liveness and safety of behaviors using it, make them easy to be understood, give a hint to system verification based on TLA.

ACKNOWLEDGMENT

We thank everyone who helped us during all researching time, especially professor Li. He is also our teacher, a man of profound learning.

REFERENCES

[1] Leslie Lamport. Specifying Systems[M]. Addison-Wesley Longman Publishing Co., Inc. 2002.

[2] Stephan Merz. Modeling and Developing Systems Using TLA+[M]. Escuela de Verano, 2005. Vol.73, Iss.3 (June 1987), pp207-244.

[3] Wolfgang Schreiner. The TLA Logic of Action I. http://www.risc.uni-linz.ac.at/people/schreine.

[4] Juntao Li, Zhengyi Tang, Xiang Li, Description and Analysis of Fairness on Temporal Logic of Actions[C], icnds, vol. 1, pp.41-44, 2009 International Conference on Networking and Digital Society, 2009

[5] Leslie Lamport. The Temporal Logic of actions [M]. ACM Transactions on Programming Languages and Systems. 1994.5,16(3):872-923.

[6] Bowen Alpern , Fred B, Schneider. Defining Liveness, Cornell University, Ithaca, NY, 1984

[7] B.Alpern, F.B.Schneider, Recognizing Safety and Liveness,TR 86-727, Computer Science Department, Cornell University, Jan 1986.

[8] S.Owicki, L.Lamport, Proving Liveness Properties of Concurrent Programs, ACM Transactions on Programming Languages and Systems 4, No.3, 1982.