A Complete Axiomatization of Propositional Projection Temporal Logic

Zhenhua Duan
Institute of Computing Theory and Technology
Xidian University, Xi'an 710071, P.R.China
zhenhua_duan@126.com

Nan Zhang
Institute of Computing Theory and Technology
Xidian University, Xi'an 710071, P.R.China
jiang-nan2006@163.com

Abstract

This paper investigates a complete axiomatic system for Propositional Projection Temporal Logic (PPTL). To this end, the syntax, semantics, and logic laws of PPTL are briefly introduced. Further, the normal form of PPTL formulas is presented. Moreover, an axiomatic system of PPTL is formalized. A set of axioms and inference rules are given in details. To assist the proof within the system, some theorems are proved by means of the axioms and rules. In addition, based on the axioms, rules and theorems, the soundness and completeness of the deductive system are proved. Finally, an example is given to illustrate how the axiom system works.

1. Introduction

Temporal logics, Linear Temporal Logic [16], Computation Tree Logic [6], Interval Temporal Logic [18, 19], Temporal Logic of Actions [13], and many others [12, 2, 24], have been proposed for specification and verification of concurrent systems for three decades. Basically, two verification approaches, model checking [5, 23] and theorem proving [3], are popular in practice. Model checking is an automatic verification approach based on model theory. The advantage of model checking is that the verification can be done automatically. However, it suffers from the state explosion problem. Also, it is less suitable for data intensive applications since the treatment of the data usually produces infinite state spaces [17]. Two successful model checking tools are SPIN [9] and SMV [17].

With theorem proving approach, to verify whether or not a system satisfies a property is to prove whether or not $\vdash \quad \rightarrow$ is a theorem within the proof system. The advantage is that theorem proving avoids the state explosion problem and can verify both finite and infinite systems, and can be done semi-automatically. It is therefore also suitable for data intensive applications. However, within the verification process, lots of assertions need to be inserted in the context of the program modeling the system, and the use of theorem prover requires considerable expertise to guide and assist the verification process. One of the famous theorem provers is PVS [20].

There are a number of proof systems and supporting tools for LTL, CTL, and TLA [10, 22, 1]. However, the expressive power of these logics is weaker than ITL which is a useful and powerful formalism for specification and verification for reactive systems since it uses a compositional operator chop (;) and an iterative operator chop-star ($*$). With ITL community, several researchers have investigated axiom systems with different extensions. Rosner and Pnueli [25] presented an axiom system for a propositional choppy logic with chop, next and until operators, and based the completeness proof on a tableau-based decision procedure. Paech [21] formalized a complete Gentzen-style proof system over finite intervals with temporal operators chop, chop-star and until. Bowman and Thompson presented a tableau-based decision procedure for PITL over finite intervals with projection. Subsequently, they presented a completeness proof for an axiomatization of this logic [4]. Moszkowski [18] presented axiom systems over finite intervals for PITL and first order ITL. The propositional part is claimed to be complete but only an outline of a proof was given. Later work extended this for projection with infinite time [19].

One of the extensions of ITL is the Projection Temporal Logic (PTL) which contains temporal operators: next and a new projection ( )[7, 8]. In this paper, with PPTL, we also extend it to contain projection-star ($\circledast$). These new operators can subsume chop , chop-star and the original projection ( ) operators. For instance (see Section 2 for details),

$$; \quad \equiv (\quad) \quad \varepsilon \quad ^* \equiv (\quad ^{\circledast}) \quad \varepsilon \text{, and}$$

$$\equiv ((\quad ^{\oplus} \quad \wedge \varepsilon) \quad (\; ; \; \wedge \varepsilon)) \wedge \quad (\;)$$

As a result, the extended logic PPTL is more expressive and represents the full regular language without loss of decidability [28]. A decision procedure for checking the satisfiability of PPTL with both finite and infinite models is

978-0-7695-3249-3/08 $25.00 © 2008 IEEE
DOI 10.1109/TASE.2008.22

given in [7, 15, 14], and based on the decision procedure, a model checking approach based on SPIN for PPTL formulas is also proposed in [27]. This enables us to verify full regular expression properties specified by PPTL formulas of concurrent systems modeled by PROMELA in SPIN as finite state programs. However, as mentioned earlier, such verification suffers from state explosion problem and is not suitable for data intensive systems. Therefore, we are motivated to formalize an axiom system for PPTL. To this end, a set of axioms and inference rules are presented; further, for convenience of proofs, a number of theorems are also proved; moreover, based on these axioms, rules, and theorems, the normal form of PPTL formulas is proved by induction on the structure of formulas; in addition, the soundness and completeness of the axiom system are proved in details.

This paper is organized as follows. In the following section, the syntax, semantics and some logic laws of PPTL are presented. The definition of the normal form of PPTL formulas is given in Section 3. In Section 4, the axiom system is formalized, in particular, axioms, inference rules and theorems are given. Then the soundness and completeness of the axiom system are proved in Section 5. An example is given in Section 6 to illustrate how the axiom system works. Finally, conclusions are drawn in Section 7.

2. Propositional Projection Temporal Logic

Our underlying logic is a Propositional Temporal Logic with projection [7, 8]. It is an extension of Propositional Interval Temporal Logic (PITL) [18, 19].

[Syntax] Let be a countable set of atomic propositions, and $_0$ non-negative integers. The formula of PPTL is given by the following grammar:

$$P ::= p \mid \bigcirc P \mid \neg P \mid P_1 \vee P_2 \mid (P_1, \ldots, P_m)\, prj\, P$$
$$\mid (P_1, \ldots, (P_i, \ldots, P_l)^{\oplus}, \ldots, P_m)\, prj\, P$$

where $\in$, $_1 \ldots \; _i \ldots \; _l \ldots \; _m (1 \leq \; \leq \; \leq$ $\in \; _0)$ and are all well-formed PPTL formulas, and $\bigcirc$, and $\oplus$ (projection-plus) are primitive temporal operators. A formula is called a state formula if it contains no temporal operators otherwise it is a temporal formula. For ease of notations, sometimes we use the abbreviation $_{(k,..,l)}$ to denote the formula sequence $_k \ldots \; _l$ ($k \leq$ $k \; \in \; _0$).

The abbreviations , f , $\wedge$, $\rightarrow$ and $\leftrightarrow$ are defined as usual. In particular, $\stackrel{\text{df}}{=} \; \vee \neg$ and $f \; \stackrel{\text{df}}{=} \; \wedge$ $\neg$ for any formula . The derived formulas are given as follows, where $\in \; _0$.

A1	$more \stackrel{\text{df}}{=} \bigcirc true$	A2	$\varepsilon \stackrel{\text{df}}{=} \neg \bigcirc true$
A3	$\bigcirc^0 P \stackrel{\text{df}}{=} P$	A4	$\bigcirc^n P \stackrel{\text{df}}{=} \bigcirc(\bigcirc^{n-1} P)(n > 0)$
A5	$\odot P \stackrel{\text{df}}{=} \varepsilon \vee \bigcirc P$	A6	$\Diamond P \stackrel{\text{df}}{=} true; P$
A7	$\Box P \stackrel{\text{df}}{=} \neg \Diamond \neg P$	A8	$P; Q \stackrel{\text{df}}{=} (P, Q)\, prj\, \varepsilon$

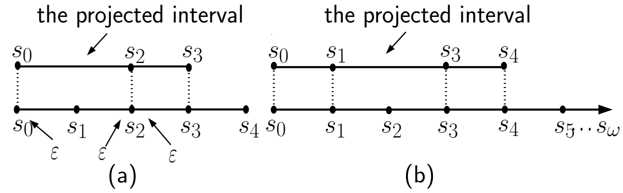

Figure 1. Projected intervals.

A9	$len(n) \stackrel{\text{df}}{=} \bigcirc^n \varepsilon$	A10	$P^+ \stackrel{\text{df}}{=} (P^{\oplus})\, prj\, \varepsilon$
A11	$skip \stackrel{\text{df}}{=} len(1)$	A12	$P^* \stackrel{\text{df}}{=} (P^{\circledast})\, prj\, \varepsilon$

A13 $(P_1, \ldots, (P_i, \ldots, P_l)^{\circledast}, \ldots, P_m)\, prj\, Q$
$\stackrel{\text{df}}{=} (P_1, \ldots, \varepsilon, \ldots, P_m)\, prj\, Q$
$\vee\, (P_1, \ldots, (P_i, \ldots, P_l)^{\oplus}, \ldots, P_m)\, prj\, Q$

A14 $(P_1, \ldots, (P_i, \ldots, P_l)^{(n)}, \ldots, P_m)\, prj\, Q \; (n > 0)$
$\stackrel{\text{df}}{=} (P_1, \ldots, \underbrace{P_i, \ldots, P_l, \ldots, P_i, \ldots, P_l}_{\text{n times}}, \ldots, P_m)\, prj\, Q$

[Semantics] State, interval, interpretation, validity, satisfiability, and precedence rules of PPTL are introduced in turn.

1. state

Following the definition of Kripke's structure [11], we define a state over to be a mapping from to $= \{ \quad f \quad \} : \; : \quad \rightarrow$. We use [ ] to denote the valuation of at state .

2. interval

An interval σ is a non-empty sequence of states, which can be finite or infinite. The length, $|\sigma|$, of σ is ω if σ is infinite, and the number of states minus 1 if σ is finite. We consider the set $_0$ of non-negative integers and ω, $_\omega = \; _0 \cup \{\omega\}$ and extend the comparison operators, $=$, , $\leq$, to $_\omega$ by considering $\omega = \omega$, and for all $\in \; _0$, ω. Furthermore, we define $\preceq$ as $\leq -\{(\omega \;\omega)\}$. To simplify definitions, we will denote σ as $_0 \cdots \; _{|\sigma|}$, where $_{|\sigma|}$ is undefined if σ is infinite. With such a notation, $\sigma_{(i \ldots j)}$ $(0 \leq \; \preceq \; \leq |\sigma|)$ denotes the sub-interval $_i \cdots \; _j$ and σ^i $(0 \leq \; \leq |\sigma|)$ denotes the prefix interval $_0 \cdots \; _i$. The concatenation of a finite σ with another interval (or empty string) σ' is denoted by $\sigma \cdot \sigma'$ (not sharing any states). Let $\sigma = \; _0 \; _1 \cdots \; _{|\sigma|}$ be an interval and $_1 \ldots \; _h$ be integers (≥ 1) such that $0 \leq \; _1 \leq \; _2 \leq \ldots \leq \; _h \preceq |\sigma|$. The projection of σ onto $_1 \ldots \; _h$ is the interval (called projected interval)

$$\sigma \downarrow (r_1, \ldots, r_h) = < s_{t_1}, s_{t_2}, \ldots, s_{t_l} >$$

where $_1 \ldots \; _l$ are obtained from $_1 \ldots \; _h$ by deleting all duplicates. That is, $_1 \ldots \; _l$ is the longest strictly increasing subsequence of $_1 \ldots \; _h$. For instance,

$$< s_0, s_1, s_2, s_3, s_4 > \downarrow (0, 0, 2, 2, 2, 3) = < s_0, s_2, s_3 >$$

this projected interval is shown in Fig. 1(a). We also need to generalize the notation of $\sigma \downarrow (\; _1 \ldots \; _h)$ to allow $_i$ to be ω. For an interval $\sigma = \; _0 \; _1 \cdots \; _{|\sigma|}$ and $0 \leq$ $_1 \leq \; _2 \leq \ldots \leq \; _h \leq |\sigma|$ ($_i \in \; _\omega$), we define

$$\sigma \downarrow (r_1, \ldots, r_h, \omega) = \sigma \downarrow (r_1, \ldots, r_h)$$

For instance,

$$< s_0, s_1, \ldots, s_\omega > \downarrow (0, 1, 3, 4, \omega, \omega) = < s_0, s_1, s_3, s_4 >$$

this projected interval is shown in Fig. 1(b).

3. interpretation

An interpretation is a triple $\mathcal{I} = (\sigma\ k\ \ )$, where σ is an interval, k integer, and an integer or ω such that $0 \leq k \preceq\ \ \leq |\sigma|$. We use the notation $(\sigma\ k\ \ ) \models$ to indicate that some formula is interpreted and satisfied over the subinterval $_k \ldots\ _j$ of σ with the current state being $_k$. The satisfaction relation ($\models$) is inductively defined in Table 1.

Table 1. Semantics

$\mathcal{I} \models p$ iff $s_k[p] = true$, for any atomic proposition p.

$\mathcal{I} \models \neg P$ iff $\mathcal{I} \not\models P$.

$\mathcal{I} \models \bigcirc P$ iff $k < j$ and $(\sigma, k+1, j) \models P$.

$\mathcal{I} \models P \vee Q$ iff $\mathcal{I} \models P$ or $\mathcal{I} \models Q$.

$\mathcal{I} \models (P_1, \ldots, P_m)\ prj\ Q$ iff there exist integers $k = r_0 \leq \cdots \leq r_{m-1} \preceq r_m \leq j$; for all $1 \leq l \leq m$, $(\sigma, r_{l-1}, r_l) \models P_l$; $\sigma' \models Q$ for one of the following σ':
- $r_m < j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_m) \cdot \sigma_{(r_m+1..j)}$, or
- $r_m = j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_h)$ for some $0 \leq h \leq m$.

$\mathcal{I} \models (P_1, \ldots, (P_i, \ldots, P_l)^{\oplus}, \ldots, P_m)\ prj\ Q$ iff one of following cases holds:
- $1 \leq i \leq l \leq m$ and there exists an integer $n \geq 1$ and $\mathcal{I} \models (P_1, \ldots, (P_i, \ldots, P_l)^{(n)}, \ldots, P_m)\ prj\ Q$, or
- $1 \leq i \leq l = m$, $j = \omega$ and there exist infinitely many integers $k = r_0 \leq r_1 \leq \cdots \leq r_k \preceq \omega$ and $\lim_{k\to\infty} r_k = \omega$ such that for all $1 \leq x \leq i-1$, $(\sigma, r_{x-1}, r_x) \models P_x$, and $(\sigma, r_{i+t(l-i+1)+n-1}, r_{i+t(l-i+1)+n}) \models P_{i+n}$, for all $t \geq 0$ and $0 \leq n \leq l-i$, and $\sigma \downarrow (r_0, r_1, \ldots, r_h, \omega) \models Q$ for some $h \in N_\omega$.

For instance, formula $(\ _1\ \ _2)$ has three possible interpretations as shown in Fig. 2(a)(b)(c). Here, and $_1$ start to be interpreted at a common state $_0$. Then $_1$ and $_2$ are interpreted sequentially. is interpreted in a parallel manner with $_1;\ _2$ over the interval, which consists of endpoints of the subintervals over which $_1\ \ _2$ are interpreted. The semantics of projection-plus ($\oplus$) is tricker. When , the last formula $_l$ of the repetition part $(\ _i \ldots\ _l)$ is not the last formula $_m$ of the formula sequence; $(\ _i \ldots\ _l)$ can be interpreted repeatedly for only finitely many times; in other words, the formulas $_{l+1} \ldots\ _m$ must be interpreted from some time point. The semantics of this case can be illustrated by the semantics of the projection. When $=$, the last formula of the repetition part is the last formula of the formula sequence. For instance, with formula $(\ _1\ (\ _2\ \ _3)^{\oplus})$, $(\ _2\ \ _3)$ can be interpreted for finitely or infinitely many times. If $(\ _2\ \ _3)$ is interpreted for infinitely many times, may terminate at some finite time point or not terminate; in other words, the projected interval over which is interpreted can be finite or infinite. The infinite case is shown in Fig. 2(d).

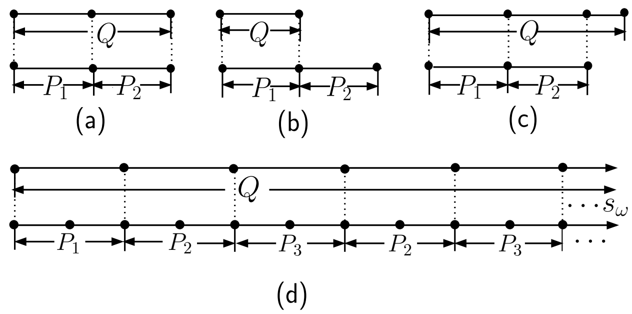

Figure 2. Possible semantics of projection and projection-plus.

4. validity and satisfiability

A formula is satisfied by an interval σ, denoted by $\sigma \models$, iff $(\sigma\ 0\ |\sigma|) \models$. A formula is called satisfiable iff $\sigma \models$ for some σ. A formula is called valid iff $\sigma \models$ for all σ, denoted by $\models$. We denote $\models \Box(\ \ \leftrightarrow\ \ )$ by $\ \equiv\ $ and $\models \Box(\ \ \rightarrow\ \ )$ by $\ \supset\ $.

Definition 1

1. A formula is called terminable iff $\ \wedge \Diamond\varepsilon \not\equiv f$.
2. A formula is called non-terminable iff $\ \equiv\ \ \wedge \Box\neg\varepsilon$.

5. precedence rules

To avoid an excessive number of parentheses, the following precedence rules are used (1=highest and 5=lowest).

1. $\neg$ 2. $\bigcirc\ \ \odot\ \ \Diamond\ \ \Box\ \ +\ \ *$ 3. $\wedge\ \ \vee$
4. $\rightarrow\ \ \leftrightarrow$ 5. $\ \ ;\ \ \oplus\ \ \circledast$

[Logic Laws] Let $_i\ \ '_i\ \ _i$ be PPTL formulas and a state formula and $_0 \stackrel{\mathrm{df}}{=} \varepsilon$. The proofs of following laws can be found in [7, 8].

L1	$\Box(P \wedge Q)$	$\equiv$	$\Box P \wedge \Box Q$	L2	$\Diamond(P \vee Q)$	$\equiv$	$\Diamond P \vee \Diamond Q$
L3	$\bigcirc(P \wedge Q)$	$\equiv$	$\bigcirc P \wedge \bigcirc Q$	L4	$\odot(P \wedge Q)$	$\equiv$	$\odot P \wedge \odot Q$
L5	$\bigcirc(P \vee Q)$	$\equiv$	$\bigcirc P \vee \bigcirc Q$	L6	$\odot(P \vee Q)$	$\equiv$	$\odot P \vee \odot Q$
L7	$\Box P$	$\equiv$	$P \wedge \odot\Box P$	L8	$\Diamond P$	$\equiv$	$P \vee \bigcirc\Diamond P$
L9	$\neg\varepsilon \wedge \neg\bigcirc P$	$\equiv$	$\neg\varepsilon \wedge \bigcirc\neg P$	L10	$\neg\odot P$	$\equiv$	$\bigcirc\neg P$
L11	$\neg\bigcirc P$	$\equiv$	$\odot\neg P$	L12	$\neg\Diamond P$	$\equiv$	$\Box\neg P$
L13	$\neg\Box P$	$\equiv$	$\Diamond\neg P$	L14	$\bigcirc P; Q$	$\equiv$	$\bigcirc(P; Q)$
L15	$\varepsilon\ prj\ Q$	$\equiv$	Q	L16	$Q\ prj\ \varepsilon$	$\equiv$	Q

L17 $(P_1, \ldots, P_m)\ prj\ \varepsilon \equiv P_1; \ldots; P_m$

L18 $(P_1, \ldots, \varepsilon \wedge w, P_i, \ldots, P_m)\ prj\ Q \equiv (P_1, \ldots, w \wedge P_i, \ldots, P_m)\ prj\ Q$

L19 $(P_1, \ldots, P_m)\ prj\ \bigcirc Q \equiv (P_1 \wedge \neg\varepsilon; ((P_2, \ldots, P_m)\ prj\ Q)) \vee (P_1 \wedge \varepsilon; ((P_2, \ldots, P_m)\ prj\ \bigcirc Q))$

L20 $(\bigcirc P_1, \ldots, P_m)\ prj\ \bigcirc Q \equiv \bigcirc(P_1; ((P_2, \ldots, P_m)\ prj\ Q))$

L21 $\bigcirc P\ prj\ \bigcirc Q \equiv \bigcirc(P; Q)$

L22 $(P_1, \ldots, (P_i \vee P_i'), \ldots, P_m)\ prj\ Q \equiv (P_1, \ldots, P_i, \ldots, P_m)\ prj\ Q \vee (P_1, \ldots, P_i', \ldots, P_m)\ prj\ Q$

L23 $(P_1, \ldots, P_m)\ prj\ (P \vee Q) \equiv (P_1, \ldots, P_m)\ prj\ P \vee (P_1, \ldots, P_m)\ prj\ Q$

L24 $(w \wedge P_1, \ldots, P_m)\ prj\ Q \equiv w \wedge (P_1, \ldots, P_m)\ prj\ Q$

L25 $(P_1, \ldots, P_m)\ prj\ (w \wedge Q) \equiv w \wedge (P_1, \ldots, P_m)\ prj\ Q$

L26 $(P_1, \ldots, P_i \wedge \Diamond\varepsilon, \ldots, P_m)\ prj\ Q \equiv$
$(P_1, \ldots, P_i, \varepsilon, \ldots, P_m)\ prj\ Q$
L27 $(P_1, \ldots, P_m)\ prj\ \varepsilon \equiv$
$(P_1, (P_2, \ldots, P_m)\ prj\ \varepsilon)\ prj\ \varepsilon \equiv$
$((P_1, \ldots, P_{m-1})\ prj\ \varepsilon, P_m)\ prj\ \varepsilon$
L28 $P \wedge \neg\Diamond\varepsilon\ prj\ Q \equiv P \wedge \neg\Diamond\varepsilon\ prj\ Q \wedge \varepsilon$
L29 $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q \equiv$
$(P_1, \ldots, P_i, \ldots, P_j, \ldots, P_m)\ prj\ Q \vee$
$(P_1, \ldots, P_i, \ldots, P_j, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q$
L30 $((P_1, \ldots, P_i)^{\oplus}, \ldots, P_m)\ prj\ Q \equiv$
$(P_1, \ldots, P_i, \ldots, P_m)\ prj\ Q \vee \bigvee_{t=1}^{i-1}$
$(\bigwedge_{h=0}^{t-1} P_h \wedge \varepsilon, P_t \wedge \neg\varepsilon, P_{(t+1,..,i)}, (P_{(1,..,i)})^{\oplus}, \ldots, P_m)\ prj\ Q \vee$
$(\bigwedge_{h=0}^{i-1} P_h \wedge \varepsilon, P_i \wedge \neg\varepsilon, (P_{(1,..,i)})^{\oplus}, \ldots, P_m)\ prj\ Q$
L31 $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ \varepsilon \equiv$
$(P_1, \ldots, (P_i; \ldots; P_j)^{+}, \ldots, P_m)\ prj\ \varepsilon$
L32 $(P_1, \ldots, (P_i, \ldots, P_m)^{\oplus}, R_1, \ldots, R_n)\ prj\ Q \equiv$
$(P_1, \ldots, (P_i, \ldots, P_m)^{\circledast}, P_i, \ldots, P_m, R_1, \ldots, R_n)\ prj\ Q$
L33 $(P_1, \ldots, (P_i, \ldots, P_m)^{\oplus})\ prj\ Q \equiv$
$((P_1, \ldots, (P_i, \ldots, P_m)^{\circledast}, P_i, \ldots, P_m)\ prj\ Q) \vee$
$((P_1, \ldots, (P_i, \ldots, P_m \wedge \Diamond\varepsilon)^{\oplus})\ prj\ Q) \wedge \neg\Diamond\varepsilon$
L34 $(P^{\oplus}, P)\ prj\ Q \supset (P, P^{\oplus})\ prj\ Q$
L35 $P^{\oplus}\ prj\ Q \equiv (P, P^{\circledast})\ prj\ Q \vee (P \wedge \neg\Diamond\varepsilon)\ prj\ Q$

3. Normal Form of PPTL

Let be a PPTL formula and $_p$ denote the set of atomic propositions appearing in . The normal form of can be defined as follows:

$$Q \equiv \bigvee_{j=1}^{n_0} (Q_j \wedge \varepsilon) \vee \bigvee_{t=1}^{n} (Q_t \wedge \bigcirc Q_t^{'}) \qquad (1)$$

where $_j \equiv \bigwedge_{k=1}^{m_0} \dot{}_{jk}$ $_t \equiv \bigwedge_{h=1}^{m} \dot{}_{th}$ $= |\ _p|$ $1 \leq$ (also $_0$) $\leq 3^l$, $1 \leq$ (also $_0$) $\leq$, $_{jk}$, $_{th} \in$ $_p$; for any $\in$ $_p$, $\dot{}$ means or $\neg$; $'_t$ is a general PPTL formula.

In some circumstances, for convenience, we write $_e \wedge \varepsilon$ instead of $\bigvee_{j=1}^{n_0}(\ _j \wedge \varepsilon)$ where $_e$ is a state formula or .
Thus,

$$Q \equiv (Q_e \wedge \varepsilon) \vee \bigvee_{i=1}^{r} (Q_i \wedge \bigcirc Q_i^{'}) \qquad (2)$$

Further, in a normal form, if $\bigvee_{i=1}^{r}\ _i \equiv$ and $\bigvee_{i \neq j}(\ _i \wedge\ _j) \equiv f$, it is called a complete normal form. The complete normal form plays an important role in transforming the negation of a formula into its normal form. For example, if formula is in its complete normal form:

$$P \equiv P_e \wedge \varepsilon \vee \bigvee_{i=1}^{r} (P_i \wedge \bigcirc P_i^{'}) \qquad (3)$$

The normal form of $\neg$ can be written as follows:

$$\neg P \equiv \neg P_e \wedge \varepsilon \vee \bigvee_{i=1}^{r} (P_i \wedge \bigcirc \neg P_i^{'}) \qquad (4)$$

In addition, any PPTL formula can be rewritten to its normal form in model theory. The proof and the algorithms transforming PPTL formulas into their normal forms and complete normal forms can be found in [7, 8]. This idea inspires us to prove that any PPTL formula can be rewritten into its normal form in our axiom system. Then, we need to consider only the normal form of formulas rather than various structures of PPTL for proving the completeness of the axiom system.

4. Axiom System Π_{pptl}

Let and be PPTL formulas. For convenience of deduction, we denote $\vdash$ $\leftrightarrow$ by $\cong$.
[Axioms] The axioms are divided into three groups w.r.t finite or infinite intervals or both, where , $'$, $_i$, $'_i$, , $'$, $_i$ are PPTL formulas, and is any state formula; $_0 \stackrel{\text{df}}{=}$ $_0 \stackrel{\text{df}}{=} \varepsilon$; $1 \leq$ $\leq$ $\leq$; $1 \leq$ $\leq$ and or ($\in$ $_0$).

Group 1: Axioms over both finite and infinite intervals

TAU $\vdash \psi$ where ψ is an instance of propositional tautologies.
NXN $\vdash \bigcirc P \rightarrow \neg \bigcirc \neg P$
NXC $\bigcirc P; Q \cong \bigcirc(P; Q)$
PNX $(\bigcirc P, P_1, \ldots, P_m)\ prj\ \bigcirc Q \cong$
$\bigcirc(P; (P_1, \ldots, P_m)\ prj\ Q)$
PDF $(P_1, \ldots, (P_i \vee P_i^{'}), \ldots, P_m)\ prj\ Q \cong$
$((P_1, \ldots, P_i, \ldots, P_m)\ prj\ Q) \vee$
$(P_1, \ldots, P_i^{'}, \ldots, P_m)\ prj\ Q$
PDB $(P_1, \ldots, P_m)\ prj\ (Q \vee Q^{'}) \cong$
$((P_1, \ldots, P_m)\ prj\ Q) \vee (P_1, \ldots, P_m)\ prj\ Q^{'}$
PSM $(P_1, \ldots, w \wedge \varepsilon, P_i, \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, w \wedge P_i, \ldots, P_m)\ prj\ Q$
PSB $(P_1, \ldots, P_m)\ prj\ (w \wedge Q) \cong w \wedge (P_1, \ldots, P_m)\ prj\ Q$
PSF $(w \wedge P_1, \ldots, P_m)\ prj\ Q \cong w \wedge (P_1, \ldots, P_m)\ prj\ Q$
PEE $(P_1, \ldots, P_i \wedge \Diamond\varepsilon, \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, P_i, \varepsilon, \ldots, P_m)\ prj\ Q$
PEC $(P_1, P_2, \ldots, P_m)\ prj\ \varepsilon \cong$
$(P_1, (P_2, \ldots, P_m)\ prj\ \varepsilon)\ prj\ \varepsilon \cong$
$((P_1, \ldots, P_{m-1})\ prj\ \varepsilon, P_m)\ prj\ \varepsilon$
PIF $P \wedge \neg\Diamond\varepsilon\ prj\ Q \cong P \wedge \neg\Diamond\varepsilon\ prj\ Q \wedge \varepsilon$
PEB $P\ prj\ \varepsilon \cong P$
PEF $\varepsilon\ prj\ P \cong P$
INX $(\bigcirc P, P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ \bigcirc Q \cong$
$\bigcirc(P; (P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q)$
IDF $(P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, (P_h \vee P_h^{'}), \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_h, \ldots, P_m)\ prj\ Q \vee$
$(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_h^{'}, \ldots, P_m)\ prj\ Q$
IDB $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ (Q \vee Q^{'}) \cong$
$((P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, P_m)\ prj\ Q) \vee$
$(P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, P_m)\ prj\ Q^{'}$
ISM $(P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, w \wedge \varepsilon, P_h, \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, w \wedge P_h, \ldots, P_m)\ prj\ Q$
ISB $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ (w \wedge Q) \cong$
$w \wedge (P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q$
ISF $(w \wedge P, P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q \cong$
$w \wedge (P, P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q$
IEE $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_h \wedge \Diamond\varepsilon, \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_h, \varepsilon, \ldots, P_m)\ prj\ Q$
IEC $(P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, P_m)\ prj\ \varepsilon \cong$
$(P_1, \ldots, (P_i; \ldots; P_j)^{+}, \ldots, P_m)\ prj\ \varepsilon$
IUP $(P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q \cong$
$(P_1, \ldots, P_i, \ldots, P_j, \ldots, P_m)\ prj\ Q \vee$
$(P_1, \ldots, P_i, \ldots, P_j, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m)\ prj\ Q$
IUM $(R_0, \ldots, R_n, (P_1, \ldots, P_i)^{\oplus}, \ldots, P_m)\ prj\ Q \cong$
$(R_0, \ldots, R_n, P_1, \ldots, P_i, \ldots, P_m)\ prj\ Q \vee \bigvee_{t=1}^{i-1}$

	$(R_{(0,..,n)}, \bigwedge_{h=0}^{t-1} P_h \wedge \varepsilon, P_t \wedge \neg\varepsilon, P_{(t+1,..,i)}, (P_{(1,..,i)})^{\oplus},$ $\ldots, P_m)\ prj\ Q \vee (R_0, \ldots, R_n, \bigwedge_{h=0}^{i-1} P_h \wedge \varepsilon, P_i \wedge \neg\varepsilon,$ $(P_{(1,..,i)})^{\oplus}, \ldots, P_m)\ prj\ Q$
IEU	$(P_1, \ldots, (P_i, \ldots, P_m)^{\oplus}, R_1, \ldots, R_n)\ prj\ Q \cong$ $(P_1, \ldots, (P_i, \ldots, P_m)^{\circledast}, P_i, \ldots, P_m, R_1, \ldots, R_n)\ prj\ Q$
IFI	$(P_1, \ldots, (P_i, \ldots, P_m)^{\oplus})\ prj\ Q \cong$ $((P_1, \ldots, (P_i, \ldots, P_m)^{\circledast}, P_i, \ldots, P_m)\ prj\ Q) \vee$ $((P_1, \ldots, (P_i, \ldots, P_m \wedge \Diamond\varepsilon)^{\oplus})\ prj\ Q) \wedge \neg\Diamond\varepsilon$
IDP	$\vdash P^+; P^+ \rightarrow P^+$

Group 2: Axioms over infinite intervals	
EEI	$\vdash \neg\Diamond\varepsilon$

Group 3: Axioms over finite intervals	
CEL	$(P_1; \bigcirc^n\varepsilon) \wedge (P_2; \bigcirc^n\varepsilon) \cong (P_1 \wedge P_2); \bigcirc^n\varepsilon$
EEF	$\vdash \Diamond\varepsilon$

[Inference Rules] In addition, the axiom system contains inference rules given in Group 4.

Group 4: Inference rules	
MP	$\vdash P \rightarrow Q,\ \vdash P \Longrightarrow\ \vdash Q$
IMP1	$\vdash P_i \rightarrow P_i^{'}\ (1 \leq i \leq m),\ \vdash Q \rightarrow Q' \Longrightarrow$ $\vdash (P_1, \ldots, P_i, \ldots, P_m)\ prj\ Q \rightarrow$ $(P_1^{'}, \ldots, P_i^{'}, \ldots, P_m^{'})\ prj\ Q'$
IMP2	$\vdash P_i \rightarrow P_i^{'}\ (1 \leq i \leq j \leq m),\ \vdash Q \rightarrow Q' \Longrightarrow$ $\vdash (P_1, \ldots, (P_{(i,..,j)})^{\oplus}, \ldots, P_m)\ prj\ Q \rightarrow$ $(P_1^{'}, \ldots, (P_{(i,..,j)}^{'})^{\oplus}, \ldots, P_m^{'})\ prj\ Q'$
ALW	$\vdash P \Longrightarrow\ \vdash \Box P$
NXT1	$\vdash P_1 \wedge \ldots \wedge P_m \rightarrow Q \Rightarrow \vdash \bigcirc P_1 \wedge \ldots \wedge \bigcirc P_m \rightarrow \bigcirc Q$
NXT2	$\vdash P \rightarrow (Q \vee \bigcirc P) \Longrightarrow\ \vdash P \rightarrow (\Diamond Q \vee \Box \bigcirc P)$

[Theorems] A set of selected theorems is given and we choose one of them to prove. The others can be proved in a similar way.

T1	$\bigcirc P \vee \bigcirc Q \cong \bigcirc(P \vee Q)$	T2	$\bigcirc P \wedge \bigcirc Q \cong \bigcirc(P \wedge Q)$
T3	$false \cong \bigcirc false$	T4	$\Diamond P \cong P \vee \bigcirc\Diamond P$
T5	$\neg \odot P \cong \bigcirc\neg P$	T6	$\Box P \cong P \wedge \bigcirc\Box P$(infinite)
T7	$\Box P \cong P \wedge \odot \Box P$	T8	$\vdash \Box P \rightarrow P$
T9	$P^*; P^* \cong P^*$		
T10	$\vdash more \rightarrow (\neg \bigcirc P \leftrightarrow \bigcirc\neg P)$		
T11	$\vdash (P^{\oplus}, P)\ prj\ Q \rightarrow (P, P^{\oplus})\ prj\ Q$		
T12	$(P_1 \wedge \varepsilon, P_2, \ldots, P_m)\ prj\ Q \cong P_1 \wedge \varepsilon; (P_{(2,..,m)})\ prj\ Q$		

PROOF OF T9

(1)	P^*	$\cong (\varepsilon\ prj\ \varepsilon) \vee P^{\oplus}\ prj\ \varepsilon$	DEF OF $\{\circledast, *\}$
(2)		$\cong \varepsilon \vee P^+$	DEF OF +
(3)	$P^*; P^*$	$\cong (\varepsilon \vee P^+, \varepsilon \vee P^+)\ prj\ \varepsilon$	(2),DEF OF ;
(4)		$\cong (\varepsilon, \varepsilon)\ prj\ \varepsilon \vee (\varepsilon, P^+)\ prj\ \varepsilon \vee$ $(P^+, \varepsilon)\ prj\ \varepsilon \vee (P^+, P^+)\ prj\ \varepsilon$	PDF
(5)		$\cong \varepsilon \vee P^+ \vee P^+ \wedge \Diamond\varepsilon \vee P^+; P^+$	PEE,PEB
(6)		$\cong P^* \vee P^+; P^+$	TAU,(2)
(7)	$\vdash P^* \rightarrow P^*; P^*$		TAU,(6)
(8)	$\vdash P^+; P^+ \rightarrow P^*$		IDP,TAU,(2)
(9)	$\vdash P^*; P^* \rightarrow P^*$		TAU,(6)(8)
(10)	$P^*; P^* \cong P^*$		(7)(9)

5. Soundness and Completeness of Π_{pptl}

Before proving the completeness of Π_{pptl}, we first consider the soundness of the axiom system.

Theorem 1 (Soundness) *The axiom system Π_{pptl} is sound, i.e. for all PPTL formula* $\ ,\ \vdash \quad \Longrightarrow \models \quad$.

Proof

It is readily to prove all the axioms are valid and all the inference rules preserve validity in model theory. The detail is omitted here. □

[Completeness] The proof of the completeness of the axiom system is based on the partition of formulas into terminable and non-terminable formulas and also on the normal form of PPTL formulas.

Basically, the normal form and complete normal form are the same as that in model theory but they are defined within the axiom system. The normal form of in Π_{pptl} can be defined as follows:

$$Q \cong Q_e \wedge \varepsilon \vee \bigvee_{t=1}^{n} (Q_t \wedge \bigcirc Q_t^{'}) \quad (5)$$

where $\ _e\ \ _t\ \ _t^{'}$ are defined in the same way as the normal form in model theory. Further, if $\bigvee_{t=1}^{n}\ \ _t \cong$ and $\bigvee_{i \neq j}(\ \ _i \wedge \ \ _j) \cong f$, the normal form is a complete normal form.

In Π_{pptl}, it is not difficult to prove the following conclusions:

① Any PPTL formula rewritten to its normal form in Π_{pptl} can be rewritten to its complete normal form in Π_{pptl}.

② If PPTL formulas $\ _1 \ldots \ \ _m$ have been rewritten to normal forms in Π_{pptl}, $(\ _1 \ldots \ \ _m)$ can be rewritten to its normal form.

③ If $\ _1 \ldots \ \ _m$ and $(\ _1 \ldots \ \ _m)$ have been rewritten to normal forms in Π_{pptl}, $(\ _1 \ldots (\ _i \ldots \ \ _l)^{\oplus} \ldots \ \ _m)$ can be rewritten to its normal form in Π_{pptl}.

Using the conclusions given above, we can prove the following theorem by induction on the syntax of PPTL.

Theorem 2 *Any PPTL formula can be rewritten to its normal form in Π_{pptl}.*

Theorem 2 tells us that any PPTL formula can be transformed into its normal form by means of axioms and inference rules. This conclusion plays an important role in the proof of completeness since we only need to consider the normal form of any formulas rather than different structures of formulas.

By the definitions of terminable and non-terminable formula given in Definition1, it is readily to prove the following facts:

Fact1 A PPTL formula is a terminable formula iff is not a non-terminable formula.

Fact2 For any PPTL formula , if is a terminable formula, is satisfiable.

From Fact2, we get to the following lemma.

Lemma 1 *For any PPTL formula , if is unsatisfiable, is not terminable.*

Lemma 2 *If a PPTL formula is non-terminable, the normal form of the formula does not contain the terminal product $_e \wedge \varepsilon$, where $_e \not\equiv f$, i.e.*

$$P \equiv \bigvee_{i=1}^{n} p_i \wedge \bigcirc P_i$$

further, every sub-formula $_i$ is non-terminable.

Proof

We prove this Lemma in two steps.
(1) The normal form of a non-terminable formula does not contain terminal product $_e \wedge \varepsilon$.
Suppose that the normal form of contains the terminal product, that is, $\equiv {}_e \wedge \varepsilon \vee \bigvee_{i=1}^{n} {}_i \wedge \bigcirc {}_i$ and $_e \not\equiv f$.

$$\begin{aligned} P \wedge \Diamond\varepsilon &\equiv (p_e \wedge \varepsilon \vee \textstyle\bigvee_{i=1}^{n} p_i \wedge \bigcirc P_i) \wedge \Diamond\varepsilon \\ &\equiv p_e \wedge \varepsilon \vee \textstyle\bigvee_{i=1}^{n} p_i \wedge \bigcirc P_i \wedge \Diamond\varepsilon \end{aligned}$$

Since $_e \not\equiv f$ and $_e$ is a state formula, we have $_e \wedge \varepsilon \not\equiv f$ and $\wedge \Diamond\varepsilon \not\equiv f$. By Definition1, we have is a terminable formula. This contradicts with the condition.
(2) Every sub-formula $_i$ is non-terminable.
From (1), we have $\equiv \bigvee_{i=1}^{n} {}_i \wedge \bigcirc {}_i$. We assume that there exists a sub-formula $_i$ being a terminable formula. Without loss of generality, let $_1$ be a terminable formula. By Definition1, we have $_1 \wedge \Diamond\varepsilon \not\equiv f$.

$$\begin{aligned} P \wedge \Diamond\varepsilon &\equiv (\textstyle\bigvee_{i=1}^{n} p_i \wedge \bigcirc P_i) \wedge \Diamond\varepsilon \\ &\equiv p_1 \wedge \bigcirc P_1 \wedge \Diamond\varepsilon \vee \textstyle\bigvee_{i=2}^{n} p_i \wedge \bigcirc P_i \wedge \Diamond\varepsilon \\ &\equiv p_1 \wedge \bigcirc (P_1 \wedge \Diamond\varepsilon) \vee \textstyle\bigvee_{i=2}^{n} p_i \wedge \bigcirc P_i \wedge \Diamond\varepsilon \end{aligned}$$

Since $_1 \wedge \Diamond\varepsilon \not\equiv f$, $_1 \wedge \bigcirc ({}_1 \wedge \Diamond\varepsilon) \not\equiv f$. Further, we have $\wedge \Diamond\varepsilon \not\equiv f$. Hence, is a terminable formula. This contradicts with the condition. □

By Lemma 2, we obtain the following corollary.

Corollary 1 *Suppose that the normal form of a PPTL formula is $_e \wedge \varepsilon \vee \bigvee_{i=1}^{n} {}_i \wedge \bigcirc {}_i$.*
(1) *If $_e \not\equiv f$ holds, is a terminable formula.*
(2) *If there exists a sub-formulas $_i$ being a terminable formula, is a terminable formula.*

By Corollary 1 and Theorem 1, it is readily to prove the following facts:
Fact3 If is non-terminable, can be transformed into the normal form without the terminal product $_e \wedge \varepsilon$ in Π_{pptl}, where $\nvdash \; _e \rightarrow f$

$$P \cong \bigvee_{i=1}^{n} p_i \wedge \bigcirc P_i$$

and for all , $_i$ is non-terminable.
Fact4 If is a state formula and $\nvdash \neg$, there exists a model $\sigma =$, such that $\sigma \models$.
Fact5 $\vdash \; \rightarrow f$, $\vdash \; \rightarrow f \Rightarrow \vdash \; \vee \; \rightarrow f$.
Fact6 $\nvdash \; \wedge \; \rightarrow f \Rightarrow \nvdash \; \rightarrow f$, $\nvdash \; \rightarrow f$.
Fact7 $\nvdash \bigcirc \; \rightarrow f \Rightarrow \nvdash \; \rightarrow f$.

In the proof of Lemma 3, we will use the fix-point theorem [26] and the fix-point induction given below [29].

Theorem 3 (Tarski's Fix-Point Theorem) *Every monotonic function over a complete lattice $\sqsubseteq$ has a unique least fix point $\sqcup_{n\in\omega} \; ^n(\bot)$ and a unique greatest fix point $\sqcap_{n\in\omega} \; ^n(\top)$. (A. Tarski 1955)*

Theorem 4 (Scott's Fix-Point Induction) *Let be a complete partial order with a bottom $(\bot)$, : $\rightarrow$ a continuous function, and D an inclusion subset of . If $\bot \in D$ and $\forall \; \in \; . \; \in D \rightarrow \; (\;) \in D$, then $f \; (\;) \in D$.*

Lemma 3 *If a PPTL formula is non-terminable and $\nvdash \; \rightarrow f$, is satisfiable.*

Proof

To prove this, we need to generate a state sequence and to prove the interval determined by the state sequence satisfies .

(1) Generating a state sequence.

Since is non-terminable, by Fact3, we have $\cong \bigvee_{i=1}^{n} {}_i \wedge \bigcirc {}_i$ and all the sub-formulas $_i$ are non-terminable. So we can repeatedly unfold formula using the normal form in Π_{pptl}. For convenience, we make some notations. Let $_i^{-1}$ denote , and k be the times of unfolding . Thus, in general, we have the following formal relation:

$$P_i^k \cong \bigvee_{i=1}^{n_{k+1}} p_i^{k+1} \wedge \bigcirc P_i^{k+1} \quad (k = -1, 0, 1, \ldots) \tag{6}$$

In following table, $_k$ denotes the set of formulas $_i^k$, obtained by the k-th unfolding $(k \geq 0)$. In particular, we choose only one formula in $_k$, $_{m_k}^k$, to generate a new set $_{k+1}$, where $1 \leq \; _k \leq \; _k$. This idea will be used in the generation of the state sequence.

k		Formula	set
-1	–	$P \cong \bigvee_{i=1}^{n_0} p_i^0 \wedge \bigcirc P_i^0$	$S_0 = \{P_i^0 \mid 1 \leq i \leq n_0\}$
0	m_0	$P_{m_0}^0 \cong \bigvee_{i=1}^{n_1} p_i^1 \wedge \bigcirc P_i^1$	$S_1 = \{P_i^1 \mid 1 \leq i \leq n_1\}$
⋮	⋮	⋮	⋮
k	m_k	$P_{m_k}^k \cong \bigvee_{i=1}^{n_{k+1}} p_i^{k+1} \wedge \bigcirc P_i^{k+1}$	$S_{k+1} = \{P_i^{k+1} \mid 1 \leq i \leq n_{k+1}\}$
⋮	⋮	⋮	⋮

The following deduction denotes a loop for generating a state sequence, where each iteration with a value of k can generate a new state $_{k+1}$ ($k = -1 \; 0 \; 1 \; \cdots \; \omega$). Since the generating process is non-terminable (because is non-terminable), we may get an infinite state sequence $\sigma =$

0 1 · · · ·

(1) $\not\vdash P_i^k \rightarrow false$	{PREMISE}
(2) $\not\vdash \bigvee_{i=1}^{n_{k+1}} p_i^{k+1} \wedge \bigcirc P_i^{k+1} \rightarrow false$	{**Fact3** }
(3) $\exists\, m_{k+1}\ 1 \leq m_{k+1} \leq n_{k+1}$ and $\not\vdash p_{m_{k+1}}^{k+1} \wedge \bigcirc P_{m_{k+1}}^{k+1} \rightarrow false$	{**Fact5**}
(4) $\not\vdash p_{m_{k+1}}^{k+1} \rightarrow false$ and $\not\vdash P_{m_{k+1}}^{k+1} \rightarrow false$	{**Fact6, Fact7**}
(5) $\exists\, s_{k+1}\ < s_{k+1} > \models p_{m_{k+1}}^{k+1}$ and $\not\vdash P_{m_{k+1}}^{k+1} \rightarrow false$	{**Fact4**}

(2) $\cong (\bigwedge_{i=0}^{k} \bigcirc^i \ {}^{i}_{m_i}) \wedge \bigcirc^{k+1} \ {}^{k}_{m_k} \vee$

It is easy to prove this conclusion by induction on k. Thus, by Theorem 1, we have $\equiv (\bigwedge_{i=0}^{k} \bigcirc^i \ {}^{i}_{m_i}) \wedge \bigcirc^{k+1} \ {}^{k}_{m_k} \vee$. In the following, we prove the infinite state sequence to be a model of $(\bigwedge_{i=0}^{\omega} \bigcirc^i \ {}^{i}_{m_i}) \wedge \bigcirc^{\omega} \ {}^{\omega}_{m_\omega}$, so it is also a model of .

(3) Each prefix of the infinite state sequence is a prefix of the final model.

The proof proceeds inductively on the length of the prefix of the state sequence.

Base: $k = 0$ $\sigma^0 =$ $_0$ $\models$ ${}^{0}_{m_0} = \bigwedge_{i=0}^{0} \bigcirc^i \ {}^{i}_{m_i}$, so the prefix of interval $_0$ is a prefix of the final model.

Induction: for $k =$ we assume that σ^n is a prefix of the final model, then we have $_{n+1}$ $\models$ ${}^{n+1}_{m_{n+1}}$ and $\sigma^n \models \bigwedge_{i=0}^{n} \bigcirc^i \ {}^{i}_{m_i}$ imply $\sigma^{n+1} \models \bigwedge_{i=0}^{n+1} \bigcirc^i \ {}^{i}_{m_i}$. So, σ^{n+1} is also a prefix of the final model.

(4) The infinite state sequence $\sigma = \sigma^\omega =$ $_0$ $_1$ $\cdots$ is the final model.

First, we make some notations. Let $\sigma_s^{-1} = \emptyset$ $\sigma_s^i = \{(0\ \ _0)\ \ldots\ (\ \ _i)\}\ (\ \in\ \ _\omega)$, where σ_s^i denotes a coded set corresponding to the prefix interval σ^i. Then we define a set $= \{\sigma_s^{-1}\ \sigma_s^0\ \sigma_s^1\ \ldots\}$ and a binary relation $\subseteq$ over , that is, $\sigma_s^i \subseteq \sigma_s^j$ iff $\leq$, and also a function $:\ \rightarrow$, $(\sigma_s^i) = \sigma_s^{i+1}$ $= -1\ 0\ 1\ \ldots$

Further, it is not hard to prove the following two conclusions: (4.1) ($\subseteq$) is a complete lattice. (4.2) is continuous. Then by Tarski's Fix-Point Theorem, we can get the least fix point of ,

$$fix(F) = \sqcup_{n \in N_\omega} F^n(\sigma_s^{-1}) = \cup_{n \in N_\omega} F^n(\sigma_s^{-1}) = \sigma_s^\omega$$

where σ_s^ω denotes the coded set corresponding to the whole state sequence. In the previous step, we have proved each prefix σ^i determined by the set σ_s^i to be a prefix of a model of . Therefore, by Scott's Fix-Point Induction, σ^ω is a model of . Thus, is satisfiable. □

Theorem 5 (Completeness) *The axiom system* Π_{pptl} *is complete, i.e. for all PPTL formula* , $\models$ $\Longrightarrow \vdash$.

Proof

$\models$		
$\Longleftrightarrow \neg$	is unsatisfiable	{**validity and satisfiability**}
$\Longleftrightarrow \neg$	is not terminable and unsatisfiable	{**Lemma 1**}
$\Longleftrightarrow \neg$	is non-terminable and unsatisfiable	{**Fact1**}
$\Longrightarrow \vdash \neg(\neg\ \ )$		{**Lemma 3**}
$\Longleftrightarrow \vdash$		{TAU}

□

6. Example

In this section, we give an example to show how the axiom system works. A requirement for a system is " is true at every even state", where is an atomic proposition. The system can be specified by the following formula,

$$((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k}\varepsilon$$

where $(\ ) \stackrel{\text{df}}{=} \Box(\varepsilon \leftrightarrow\ )$. We want to prove the system satisfies the property $\bigwedge_{m=0}^{k} \bigcirc^{2m}$, that is,

$$\vdash ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k}\varepsilon \rightarrow \bigwedge_{m=0}^{k} \bigcirc^{2m} p$$

First, it is readily to prove the following theorems in Π_{pptl}.

ET1 $\Box p; r \wedge \varepsilon \cong p \wedge (\odot \Box p; r \wedge \varepsilon)$
ET2 $r \wedge halt(r) \cong r \wedge \varepsilon$
ET3 $((\varepsilon, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc P \cong false$
ET4 $\Box p; r \wedge \varepsilon \cong p \wedge r \wedge \varepsilon \vee p \wedge \bigcirc(\Box p; r \wedge \varepsilon)$
ET5 $\vdash \bigcirc^2 P \wedge halt(r) \rightarrow \bigcirc^2(P \wedge halt(r))$

Proof

The proof proceeds inductively on k.

Base: $k = 0$

(1) $((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \varepsilon$
$\cong ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (p \wedge (\odot \Box p; r \wedge \varepsilon))) \wedge halt(r) \wedge \varepsilon$
{ET1}
(2) $\cong p \wedge ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\odot \Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \varepsilon$
{ISB}
(3) $\vdash ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \varepsilon \rightarrow p$
{TAU}

Induction: Suppose for all $k \geq 0$, the conclusion holds. Then for $k + 1$,

(1) $((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon$
$\cong ((\bigcirc^2\varepsilon^{\oplus}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon$
{DEF OF $\circledast$, ET3, TAU}
(2) $\cong ((\bigcirc^2\varepsilon, \varepsilon, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon \vee$
$((\bigcirc^2\varepsilon, \bigcirc^2\varepsilon^{\oplus}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge$
$\bigcirc^{2k+2}\varepsilon$ {IUP,ISM}
(3) $\cong p \wedge r \wedge ((\bigcirc^2\varepsilon, \bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ \varepsilon) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon$
$\vee p \wedge ((\bigcirc^2\varepsilon, \bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ \bigcirc (\Box p; r \wedge \varepsilon)) \wedge halt(r)$
$\wedge \bigcirc^{2k+2}\varepsilon$ {DEF OF $\circledast$, ET4, IDB, ISB}
(4) $\cong p \wedge ((\bigcirc^2\varepsilon, \bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ \bigcirc (\Box p; r \wedge \varepsilon)) \wedge halt(r)$
$\wedge \bigcirc^{2k+2}\varepsilon$ {ET2, TAU}
(5) $\cong p \wedge \bigcirc^2((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge$
$\bigcirc^{2k+2}\varepsilon$ {INX, NXC, PSM, PEB}
(6) $\vdash ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon$
$\rightarrow p \wedge \bigcirc^2(((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r)$
$\wedge \bigcirc^{2k+2}\varepsilon)$ {T2, ET5, (5)}
(7) $\vdash ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k}\varepsilon \rightarrow$
$\bigwedge_{m=0}^{k} \bigcirc^{2m} p$ {HYPOTHESIS}

(8) $\vdash ((\bigcirc^2\varepsilon^{\circledast}, r \wedge \varepsilon)\ prj\ (\Box p; r \wedge \varepsilon)) \wedge halt(r) \wedge \bigcirc^{2k+2}\varepsilon)$
$\rightarrow \bigwedge_{m=0}^{k+1} \bigcirc^{2m} p$ {NXT1, TAU}

So the property also holds on $k+1$. Thus, the system satisfies the property. □

7. Conclusion

In this paper, we presented a complete axiom system for PPTL which supports both finite and infinite models. We also proved the soundness and completeness of the axiom system. Further, an example was given to illustrate how the system works. This enables us to verify properties of systems by means of the deductive approach. However, in order to verify properties of a real system, a theorem prover is required. Therefore, we have developed a theorem prover based on PVS to support automatic verification. It is merely a prototype and lots of efforts are needed to improve it. Moreover, to examine the axiomatic system further, several case studies with larger examples are also required.

In addition, as practical applications, we will further investigate verification techniques for composite web-services based on PVS using PPTL since data flow is intensively involved with the composition process and the Model Checking approach might be unsuitable. To do so, lots of research work are needed, and we are motivated to formalize some useful verification techniques using PPTL in this area in the future.

References

[1] M. Abadi. An axiomatization of Lamport's temporal logic of actions. *LNCS*, 458:57–69, 1990.

[2] B. Banieqbal and H. Barringer. Temporal logic with fixed points. *LNCS*, 389:62–74, 1987.

[3] W. Bledsoe and D. Loveland. *Automating Theorem Proving: After 25 Years*. Amer Mathematical Society, USA, 1984.

[4] H. Bowman and S. Thompson. A decision procedure and complete axiomatization of finite interval logic with projection. *Journal of Logic and Computation*, 13:195–239, 2003.

[5] E. Clarke and E. Emerson. Design and synthesis of synchronisation skeletons using branching time temporal logic. *LNCS*, 131:52–71, 1981.

[6] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite state concurrent system using temporal logic specification. *ACM Trans. on Programming Languages and Systems*, 8(2):244–263, 1986.

[7] Z. Duan, C. Tian, and L. Zhang. A decision procedure for propositional projection temporal logic. *Acta Informatica*, 45:43–78, 2008.

[8] Z. Duan, X. Yang, and M. Koutny. Frammed temporal logic programming. *Science of Computer Programming*, 70:31–61, 2008.

[9] G. Holzmann. The model checker SPIN. *IEEE Trans. on Software Engineering*, 23(5):279–295, 1997.

[10] Y. Kesten and A. Pnueli. A complete proof system for QPTL. *In: Proceedings of the 10th IEEE Symposium on Logic in Computer Science*, pages 2–12, 1995.

[11] S. Kripke. Semantical analysis of modal logic I: normal propositional calculi. *Z. Math. Logik Grund. Math.*, 9:67–96, 1963.

[12] F. Kröger. *Temporal Logic of Programs*. Springer-Verlag, 1987.

[13] L. Lamport. The temporal logic of actions. *ACM TOPLAS*, 16:872–923, 1994.

[14] S. Liu and Y. Chen. A relation-based method combining functional and structural testing for test case generation. *Journal of Systems and Software*, 81:234–248, 2008.

[15] S. Liu and H. Wang. An automated approach to specification animation for validation. *Journal of Systems and Software*, 80:1271–1285, 2007.

[16] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent System*. Springer-Verlag, New York, 1992.

[17] K. McMillan. *Symbolic Model Checking: An Approach to the State Explosion Problem*. Kluwer Academic Publisher, Dordrecht, 1993.

[18] B. Moszkowski. Some very compositional temporal properties. *In: Proceedings of Programming Concepts, Methods and Calculi, IFIP Trans.*, A-56:307–326, 1994.

[19] B. Moszkowski. Compositional reasoning about projected and infinite time. *In: Proceedings of the 1st IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'95)*, pages 238–245, 1995.

[20] S. Owre, J. Rushby, and N. Shankar. PVS: A prototype verification system. *In: Proceedings of the 11th International Conference on Automated Deduction (CADE), LNAI*, 607:748–752, 1992.

[21] B. Paech. Gentzen-Systems for propositional temporal logics. *In: Proceedings of the 2nd Workshop on Computer Science Logic, Duisburg (FRG), LNCS*, 385:240–253, 1988.

[22] A. Pnueli and Y. Kesten. A deductive proof system for CTL. *In: Proceedings of the 13th International Conference on Concurrency Theory*, pages 24–40, 2002.

[23] J. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. *In: Proceedings of the 5th International Symposium in Programming, LNCS*, 137:337–351, 1982.

[24] N. Rescher and A. Urquhart. *Temporal Logic*. Springer-Verlag, New York, 1978.

[25] R. Rosner and A. Pnueli. A choppy logic. *In: Proceedings of 1st IEEE Symposium on Logic in Computer Science*, pages 306–314, 1986.

[26] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math*, 5:285–309, 1955.

[27] C. Tian and Z. Duan. Model checking propositional projection temporal logic based on SPIN. *In: Proceedings of the 9th International Annual Conf. on Formal Engineering Methods, LNCS*, 4789:246–265, 2007.

[28] C. Tian and Z. Duan. Propositional projection temporal logic, Buchi automata and ω-regular expressions. *To appear in: Proceedings of the 5th International Annual Conf. on Thoery and Applications of Models of Computation (TAMC 2008), LNCS*, 4978, 2008.

[29] G. Winskel. *The Formal Semantics of Programming Languages*. Foundations of Computing, MIT, USA, 1993.