Theoretical Computer Science 609 (2016) 639–657

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A complete axiom system for propositional projection temporal logic with cylinder computation model ☆

Nan Zhang, Zhenhua Duan *, Cong Tian

Institute of Computing Theory and Technology, and ISN Laboratory, Xidian University, Xi'an 710071, PR China

ARTICLE INFO

Article history:
Received 29 October 2014
Received in revised form 9 April 2015
Accepted 6 May 2015
Available online 13 May 2015

Keywords:
Axiom system
Multi-core
Model
Specification
Verification

ABSTRACT

To specify and verify multi-core parallel programs in a uniform framework, this paper proposes an axiom system for CCM–PPTL which extends that of PPTL by including transformation rules for sequence expressions and axioms as well as inference rules on the CCM construct. Further, the soundness and completeness of the extended axiom system are proved.

1. Introduction

With the rapid development of integrated circuits technology and the demand for higher performance, on-chip multi-core processors (CMP) have been brought into being. The reality of multi-core processor has made parallel programs pervasive. Creating a correct parallel program is not a straightforward process even for a considerable small program, because programmers are forced to consider that the program will always yield to a correct result no matter in what order the instructions are executed. To improve the reliability of parallel programs, formal verification is an important viable approach. Modeling multi-core parallel programs is a crucial step for formal verification of correctness and reliability of multi-core parallel programs.

Model checking [5,18] and theorem proving [3] are two key verification methods. With model checking, the system is often modeled as a finite transition system or automaton M, and the property is specified using a temporal logic formula P. Then a model checking procedure is employed to check whether or not $M \models P$ is valid. If so, the property is verified otherwise a counterexample can be found. The advantage of model checking is that the verification can be done automatically. However, model checking suffers from the state explosion problem [14]. Further, most of web applications are data-intensive which are not suitable to be verified by means of model checking since the treatment of the data usually leads to a huge, even infinite state space. Some successful model checking tools are SPIN [13], SMV [14] and so on. By contrast, theorem proving can handle many complex structures abstractly without suffering from state space explosion problem. However, theorem proving requires more human interventions and is often time-consuming. With theorem proving, both the system behavior and the desired property are specified as formulas, say S and P, in some appropriate logic. To prove that the

☆ This research is supported by the NSFC under Grant Nos. 61133001, 61202038, 61322202, 61420106004 and 91418201.
* Corresponding author.
E-mail addresses: nanzhang@xidian.edu.cn (N. Zhang), zhhduan@mail.xidian.edu.cn (Z. Duan), c.tian@mail.xidian.edu.cn (C. Tian).

http://dx.doi.org/10.1016/j.tcs.2015.05.007
0304-3975/

system satisfies the property amounts to proving that $\vdash S \rightarrow P$ is a theorem within the proof system of the logic. Some famous theorem provers are PVS [16], ACL2 [4], Coq [2], Isabella [17], HOL [11] and so on.

Verification of multi-core parallel programs raises a great challenge for theorem proving since it requires that the logic for modeling multi-core systems and specifying the expected properties has a powerful expressiveness. However, the widely used Propositional Linear Temporal Logic (PLTL) and Computational Tree Logic (CTL) are not powerful enough. In fact, they are not full regular. Although Quantified Linear time Temporal Logic (QLTL) [19], Extended Temporal Logic (ETL) [22] and linear μ-calculus [21] have full regular expressiveness, these logics are not intuitive to use in practice. Propositional Projection Temporal Logic (PPTL) [6] allows us to specify ω full regular properties [20]. Further, a decision procedure [9,8] and a complete proof system for PPTL [10] have been established. A model checker based on SPIN [7] and a theorem prover based on PVS have also been developed. Cylinder Computation Model (CCM) [23] is a concurrent semantic model which is defined based on PPTL and has been implemented in the interpreter of MSVL (Modeling, Simulation and Verification Language) [6], which is an executable subset of Projection Temporal Logic. CCM can be employed to model multi-core parallel programs since the sequence expressions in it have the nature of regular expressions. With CCM, the autonomy and parallelism of the processes occupying different cores on one chip can be described neatly and concisely. In [10], we have proposed an axiom system for PPTL, and proved its soundness and completeness. To specify and verify multi-core parallel programs in a uniform framework, this paper proposes an axiom system for CCM–PPTL which extends that of PPTL by including transformation rules for sequence expressions and axioms as well as inference rules on the CCM construct. Furthermore, the soundness and completeness of the extended axiom system are also proved.

The paper is organized as follows: in the next section, the underlying logic PPTL and CCM are reviewed. Based on PPTL, CCM–PPTL is introduced in Section 3, including the syntax, semantics and some logical laws regarding the CCM construct. In Section 4, an axiom system for CCM–PPTL is formalized. In particular, the axioms, inference rules and the proofs of the soundness and completeness are given in detail. In Section 5, an example is given to illustrate how to use CCM and its proof system to model and verify practical algorithms. Finally, conclusions are drawn in Section 6.

2. Preliminaries

Our underlying logic is Propositional Projection Temporal Logic (PPTL), which is an interval-based temporal logic with ω full regular expressiveness. For more details on PPTL, refer to [6,10].

2.1. Propositional projection temporal logic

2.1.1. Syntax

The formula P of PPTL can be defined 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 $p \in Prop$, $P_1, \ldots, P_i, \ldots, P_l, \ldots, P_m$ $(1 \leq i \leq l \leq m,\ i, l, m \in N_0)$ and P are all well-formed PPTL formulas, and $\bigcirc$, prj and $prj\oplus$ (projection-plus) are primitive temporal operators. A formula is called a state formula if it contains no temporal operators, otherwise it is called a temporal formula.

2.1.2. Semantics

We define a state s over $Prop$ to be a mapping from $Prop$ to B.

$$s : Prop \rightarrow B$$

We use $s[p]$ to denote the valuation of p at state s.

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 N_0 of non-negative integers and ω, $N_\omega = N_0 \cup \{\omega\}$ and extend the comparison operators, $=$, $<$, $\leq$, to N_ω by considering $\omega = \omega$, and for all $i \in N_0$, $i < \omega$. Furthermore, we define $\preceq$ as $\leq -\{(\omega, \omega)\}$. To simplify definitions, we will denote σ as $\langle s_0, \ldots, s_{|\sigma|}\rangle$, where $s_{|\sigma|}$ is undefined if σ is infinite. With such a notation, $\sigma_{(i\ldots j)} (0 \leq i \preceq j \leq |\sigma|)$ denotes the sub-interval $\langle s_i, \ldots, s_j\rangle$ and σ^i $(0 \leq i \leq |\sigma|)$ denotes the prefix interval $\langle s_0, \ldots, s_i\rangle$. The concatenation of a finite σ with another interval (or empty string) σ' is denoted by $\sigma \bullet \sigma'$ (not sharing any states). Let $\sigma = \langle s_0, s_1, \ldots, s_{|\sigma|}\rangle$ be an interval and $r_1, \ldots, r_h$ be integers $(h \geq 1)$ such that $0 \leq r_1 \leq r_2 \leq \ldots \leq r_h \preceq |\sigma|$. The projection of σ onto $r_1, \ldots, r_h$ is the interval (called projected interval)

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

where $t_1, \ldots, t_l$ are obtained from $r_1, \ldots, r_h$ by deleting all duplicates. That is, $t_1, \ldots, t_l$ is the longest strictly increasing subsequence of $r_1, \ldots, r_h$.

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

Table 1
Semantics of PPTL.

1. $\mathcal{I} \models$	p	iff	$s_k[p] = true$, for any atomic proposition p.
2. $\mathcal{I} \models$	$\neg P$	iff	$\mathcal{I} \not\models P$.
3. $\mathcal{I} \models$	$\bigcirc P$	iff	$k < j$ and $(\sigma, k+1, j) \models P$.
4. $\mathcal{I} \models$	$P \vee Q$	iff	$\mathcal{I} \models P$ or $\mathcal{I} \models Q$.
5. $\mathcal{I} \models$	$(P_1, \ldots, P_m)$ *prj* Q	iff	there exist integers $k = r_0 \leq \cdots \leq r_{m-1} \preceq r_m \leq j$; for all $1 \leq l \leq m$, $(\sigma, r_{l-1}, r_l) \models P_l$; $(\sigma', 0, \lvert\sigma'\rvert) \models Q$ for one of the following σ': (a) $r_m < j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_m) \bullet \sigma_{(r_m+1..j)}$, or (b) $r_m = j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_h)$ for some $0 \leq h \leq m$.
6. $\mathcal{I} \models$	$(P_1, \ldots, (P_u, \ldots, P_l)^{\oplus}, \ldots, P_m)$ *prj* Q	iff	one of following cases holds: (a) $1 \leq u \leq l \leq m$ and there exists an integer $n \geq 1$ and $\mathcal{I} \models (P_1, \ldots, (P_u, \ldots, P_l)^{(n)}, \ldots, P_m)$ *prj* Q, or (b) $1 \leq u \leq l = m$, $j = \omega$ and there exist infinitely many integers $k = r_0 \leq r_1 \leq \cdots \leq r_n \preceq \omega$ and $\lim_{n\to\infty} r_n = \omega$ such that for all $1 \leq x \leq u-1$, $(\sigma, r_{x-1}, r_x) \models P_x$, and $(\sigma, r_{u+t(l-u+1)+n-1}, r_{u+t(l-u+1)+n}) \models P_{u+n}$, for all $t \geq 0$ and $0 \leq n \leq l-u$, and $\sigma' = \sigma \downarrow (r_0, r_1, \ldots, r_h, \omega)$ and $(\sigma', 0, \lvert\sigma'\rvert) \models Q$ for some $h \in N_\omega$.

Table 2
Abbreviations.

A1	more	$\stackrel{\text{def}}{=}$	$\bigcirc$true	A2	ε	$\stackrel{\text{def}}{=}$	$\neg \bigcirc$ true
A3	$\bigcirc^0 P$	$\stackrel{\text{def}}{=}$	P	A4	$\bigcirc^n P$	$\stackrel{\text{def}}{=}$	$\bigcirc(\bigcirc^{n-1} P)$
A5	$P; Q$	$\stackrel{\text{def}}{=}$	(P, Q) *prj* ε	A6	$\Diamond P$	$\stackrel{\text{def}}{=}$	true; P
A7	$\Box P$	$\stackrel{\text{def}}{=}$	$\neg\Diamond\neg P$	A8	$\odot P$	$\stackrel{\text{def}}{=}$	$\varepsilon \vee \bigcirc P$
A9	len(n)	$\stackrel{\text{def}}{=}$	$\bigcirc^n \varepsilon$	A10	P^+	$\stackrel{\text{def}}{=}$	$(P^{\oplus})$ *prj* ε
A11	P^*	$\stackrel{\text{def}}{=}$	$\varepsilon \vee P^+$	A12	fin(P)	$\stackrel{\text{def}}{=}$	$\Box(\varepsilon \rightarrow P)$
A13	$P_1 \| P_2$	$\stackrel{\text{def}}{=}$	$P_1 \wedge (P_2; \text{true}) \vee P_2 \wedge (P_1; \text{true})$				

2.1.3. Abbreviations

The abbreviations true, false, $\wedge$, $\rightarrow$ and $\leftrightarrow$ are defined as usual. In particular, true $\stackrel{\text{def}}{=} P \vee \neg P$ and false $\stackrel{\text{def}}{=} P \wedge \neg P$ for any formula P. We also use the abbreviations given in Table 2.

2.1.4. Axiom system of PPTL

The details of the proof system Π_{pptl} is presented in [10]. For the convenience of deduction, we will denote $\vdash P \leftrightarrow Q$ by $P \cong Q$ and $\vdash P \rightarrow Q$ by $P \supset Q$. Moreover, for the ease of presentation, we will use:

- τ to denote an empty formula sequence which does not contains any formulas;
- $P[i, n]$ to denote a formula sequence $P_i, \ldots, P_n$ where $i \leq n$; when $i = n$, $P[i, i]$ denotes P_i;
- $P[i, n, \tau]$ to denote $P[i, n]$ or τ;
- $P[i, n][j, l]^0$ to denote $P[i, n]$;
- $P[i, n][j, l]^1$ to denote $P_i, \ldots, (P_j, \ldots, P_l)^{\oplus}, \ldots, P_n$ where $i \leq j \leq l \leq n$; when $j = i$ and $l = n$, $P[i, n][j, l]^1$ denotes $(P_i, \ldots, P_n)^{\oplus}$; further, when $i = j = l = n$, $P[i, i][i, i]^1$ denotes $(P_i)^{\oplus}$;
- $P[i, n][j, l]^{-1}$ to denote τ;
- $P[i, n][j, l]$ to denote $P[i, n][j, l]^0$ or $P[i, n][j, l]^1$;
- $P[i, n, \tau][j, l]$ to denote $P[i, n][j, l]^1$ or $P[i, n]$ or τ;

In addition, we define (τ) *prj* P to be P. Any formula sequence $P[i, n, \tau][j, l]$ joined with τ by ',' is still itself. That is, we have:

$$\underline{P[i, n, \tau][j, l], \tau} = \underline{\tau, P[i, n, \tau][j, l]} = P[i, n, \tau][j, l]$$

The set of axioms of Π_{pptl} is given in Table 3, where w is a state formula and $m \geq 1$.

The inference rules are given in Table 4. The proofs of soundness and completeness are given in detail in [10]. The following are two useful conclusions.

Conclusion 1 *(Soundness).* For any PPTL formula P, if $\vdash P$, then $\models P$.

Conclusion 2 *(Completeness).* For any PPTL formula P, if $\models P$, then $\vdash P$.

2.2. Cylinder computation model

In this section, the details of Cylinder Computation Model (CCM) is introduced [23], including its syntax and semantics which are defined based on sequence expressions.

Table 3
Axioms.

TAU	$\vdash \psi$ where ψ is an instance of a propositional tautology
NXN	$\vdash \bigcirc P \to \neg \bigcirc \neg P$
ECI	$\vdash P \wedge \Diamond \neg P \to \Diamond (P \wedge \bigcirc \neg P)$
TRU	$\vdash (\bigcirc \varepsilon)^*$
AES	$\Box w \cong w \wedge (\bigcirc (w \wedge \varepsilon))^*$
AIS	$\vdash P \wedge \Box (P \to Q \vee w \wedge \bigcirc^n P)$
	$\to Q \vee w \wedge (\bigcirc^n (w \wedge \varepsilon))^+ \wedge inf \vee w \wedge (\bigcirc^n (w \wedge \varepsilon))^*; \bigcirc^n Q$
CEL	$(P_1; P_3 \wedge \bigcirc^n \varepsilon) \wedge (P_2; P_4 \wedge \bigcirc^n \varepsilon) \cong (P_1 \wedge P_2); (P_3 \wedge P_4 \wedge \bigcirc^n \varepsilon)$
CDP	$\vdash P^+; P^+ \to P^+$
PEB	$P \ prj \ \varepsilon \cong P$
PEF	$\varepsilon \ prj \ P \cong P$
PNX	$(\bigcirc P, P[i, n, \tau][j, l]) \ prj \ \bigcirc Q \cong \bigcirc (P; (P[i, n, \tau][j, l]) \ prj \ Q)$
PDF	$(P[1, n_1][j_1, l_1]^i, (P' \vee P''), Q[1, n_2][j_2, l_2]^j) \ prj \ R$
	$\cong (P[1, n_1][j_1, l_1]^i, P', Q[1, n_2][j_2, l_2]^j) \ prj \ R \vee$
	$(P[1, n_1][j_1, l_1]^i, P'', Q[1, n_2][j_2, l_2]^j) \ prj \ R \quad (i + j \in \{-2, -1, 0, 1\})$
PDB	$(P[1, n][j, l]) \ prj \ (Q \vee Q') \cong ((P[1, n][j, l]) \ prj \ Q) \vee ((P[1, n][j, l]) \ prj \ Q')$
PSM	$(P[1, n_1][j_1, l_1]^i, w \wedge \varepsilon, P', Q[1, n_2][j_2, l_2]^j) \ prj \ R$
	$\cong (P[1, n_1][j_1, l_1]^i, w \wedge P', Q[1, n_2][j_2, l_2]^j) \ prj \ R \quad (i + j \in \{-2, -1, 0, 1\})$
PSB	$(P[1, n][j, l]) \ prj \ (w \wedge Q) \cong w \wedge (P[1, n][j, l]) \ prj \ Q$
PSF	$(w \wedge P', P[1, n, \tau][j, l]) \ prj \ Q \cong w \wedge (P', P[1, n, \tau][j, l]) \ prj \ Q$
PEE	$(P[1, n_1][j_1, l_1]^i, P' \wedge \Diamond \varepsilon, Q[1, n_2][j_2, l_2]^j) \ prj \ R$
	$\cong (P[1, n_1][j_1, l_1]^i, P', \varepsilon, Q[1, n_2][j_2, l_2]^j) \ prj \ R \quad (i + j \in \{-2, -1, 0, 1\})$
PEC	$(P', P[1, n, \tau][j, l], P'') \ prj \ \varepsilon \cong (P', (P[1, n, \tau][j, l], P'') \ prj \ \varepsilon) \ prj \ \varepsilon$
	$\cong ((P', P[1, n, \tau][j, l]) \ prj \ \varepsilon, P'') \ prj \ \varepsilon$
PIF	$P \wedge \neg \Diamond \varepsilon \ prj \ Q \cong P \wedge \neg \Diamond \varepsilon \ prj \ Q \wedge \varepsilon$
IEC	$(P[1, n_1, \tau], Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ \varepsilon$
	$\cong (P[1, n_1, \tau], (Q_1; \ldots; Q_{n_2})^+, R[1, n_3, \tau]) \ prj \ \varepsilon$
IUP	$(P[1, n_1, \tau], Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ P'$
	$\cong (P[1, n_1, \tau], Q[1, n_2], R[1, n_3, \tau]) \ prj \ P' \vee$
	$(P[1, n_1, \tau], Q[1, n_2], Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ P'$
IUM	$(P[1, n_1, \tau], Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ P'$
	$\cong (P[1, n_1, \tau], Q[1, n_2], R[1, n_3, \tau]) \ prj \ P'$
	$\vee \bigvee_{t=1}^{n_2 - 1} (P[1, n_1, \tau], \bigwedge_{h=0}^{t-1} Q_h \wedge \varepsilon, Q_t \wedge \neg \varepsilon, Q[t+1, n_2], Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ P'$
	$\vee (P[1, n_1, \tau], \bigwedge_{h=0}^{m-1} Q_h \wedge \varepsilon, Q_{n_2} \wedge \neg \varepsilon, Q[1, n_2][1, n_2]^1, R[1, n_3, \tau]) \ prj \ P'$
IEU	$(P[1, n_1, \tau], Q[1, n_2][1, n_2]^1, R[1, n_3]) \ prj \ P'$
	$\cong (P[1, n_1, \tau], Q[1, n_2, \tau][1, n_2], Q[1, n_2], R[1, n_3]) \ prj \ P'$
IFI	$(P[1, n_1, \tau], (Q[1, n_2, \tau], R)^{\oplus}) \ prj \ P'$
	$\cong (P[1, n_1, \tau], (Q[1, n_2, \tau], R)^{\circledast}, Q[1, n_2, \tau], R) \ prj \ P'$
	$\vee ((P[1, n_1, \tau], (Q[1, n_2, \tau], R \wedge \Diamond \varepsilon)^{\oplus}) \ prj \ P') \wedge \neg \Diamond \varepsilon$
IOC	$\vdash ((P)^{\circledast}) \ prj \ (Q; Q') \to (((P)^{\circledast}) \ prj \ Q); (((P)^{\circledast}) \ prj \ Q')$

Table 4
Inference rules.

MP	$\vdash P \to Q, \ \vdash P \Longrightarrow \ \vdash Q$
IMP1	$\vdash P_i \to P_i' \ (1 \le i \le m), \ \vdash Q \to Q' \Longrightarrow$
	$\vdash (P_1, \ldots, P_i, \ldots, P_m) \ prj \ Q \to (P_1', \ldots, P_i', \ldots, P_m') \ prj \ Q'$
IMP2	$\vdash P_i \to P_i' \ (1 \le i \le j \le m), \ \vdash Q \to Q' \Longrightarrow$
	$\vdash (P_1, \ldots, (P_i, \ldots, P_j)^{\oplus}, \ldots, P_m) \ prj \ Q \to (P_1', \ldots, (P_i', \ldots, P_j')^{\oplus}, \ldots, P_m') \ prj \ Q'$
ALW	$\vdash P \Longrightarrow \ \vdash \Box P$
NXT1	$\vdash P_1 \wedge \ldots \wedge P_m \to Q \Longrightarrow \ \vdash \bigcirc P_1 \wedge \ldots \wedge \bigcirc P_m \to \bigcirc Q$
NXT2	$\vdash P \to (Q \vee \bigcirc P) \Longrightarrow \ \vdash P \to (\Diamond Q \vee \Box \bigcirc P)$

2.2.1. Sequence Expression

Syntax

$$l ::= \emptyset \mid \epsilon \mid n \mid l_1 \cdot l_2 \mid l_1 \otimes l_2 \mid l^*$$

From the syntax, we can see that the sequence expression is an analogue of regular expressions where $\emptyset$ denotes the empty set, ϵ empty sequence expression and $n \in N_0$ single element expression. The concatenation ("$\cdot$"), sum ("$\otimes$") of any two sequence expressions, or Kleene closure ("$*$") of a sequence expression is also a sequence expression.

Semantics The semantics of sequence expressions can be defined by a satisfaction relation, $\Vdash$, by means of interpretation $\mathcal{I} = (\sigma, k, j)$, given in Table 5.

Table 5
Semantics of sequence expressions.

1.	$\mathcal{I} \not\Vdash$	$\emptyset$	for all $\mathcal{I}$.	
2.	$\mathcal{I} \Vdash$	ϵ	iff	$j = k$.
3.	$\mathcal{I} \Vdash$	n	iff	$j - k = n$.
4.	$\mathcal{I} \Vdash$	$l_1 \cdot l_2$	iff	there exists r, $k \leq r \preceq j$, such that $\mathcal{I}_1 = (\sigma, k, r) \Vdash l_1$ and $\mathcal{I}_2 = (\sigma, r, j) \Vdash l_2$.
5.	$\mathcal{I} \Vdash$	$l_1 \otimes l_2$	iff	$\mathcal{I} \Vdash l_1$ or $\mathcal{I} \Vdash l_2$.
6.	$\mathcal{I} \Vdash$	l^*	iff	$j = k$ or there exist finitely many integers $k = r_0 \leq r_1 \leq \ldots \preceq r_n = j$ such that for all h, $1 \leq h \leq n$, $(\sigma, r_{h-1}, r_h) \Vdash l$.

Table 6
Algebraic laws of sequence expressions.

R1	$\epsilon = 0$	R11	$l \otimes l = l$
R2	$l \otimes \emptyset = \emptyset \otimes l = l$	R12	$l_1 \otimes l_2 = l_2 \otimes l_1$
R3	$0 \cdot l = l \cdot 0 = l$	R13	$l_1 \cdot (l_2 \otimes l_3) = (l_1 \cdot l_2) \otimes (l_1 \cdot l_3)$
R4	$\emptyset \cdot l = l \cdot \emptyset = \emptyset$	R14	$(l_1 \otimes l_2) \cdot l_3 = (l_1 \cdot l_3) \otimes (l_2 \cdot l_3)$
R5	$l \cdot l^* = l^* \cdot l$	R15	$l_1 \cdot (l_2 \cdot l_1)^* = (l_1 \cdot l_2)^* \cdot l_1$
R6	$l^* \cdot l^* = l^*$	R16	$l_1 \cdot (l_2 \cdot l_3) = (l_1 \cdot l_2) \cdot l_3$
R7	$(l^*)^* = l^*$	R17	$l_1 \otimes (l_2 \otimes l_3) = (l_1 \otimes l_2) \otimes l_3$
R8	$l^* = \epsilon \otimes (l \cdot l^*)$	R18	$\emptyset^* = \epsilon$
R9	$0^* = 0$	R19	$(0 \otimes l)^* = l^*$
R10	If $l = (l_1 \cdot l) \otimes l_2$, then $l = l_1^* \cdot l_2$ (Arden's Rule [1,15])		

In the semantics of sequence expressions, $\emptyset$ cannot be satisfied by any interpretation. The empty sequence expression ϵ is equivalent to the sequence expression 0. In order to avoid an excessive number of parentheses, the precedence rules are given from high to low: (1) $*$ (iteration); (2) $\cdot$ (concatenation); (3) $\otimes$ (selection).

For any two sequence expressions l_1 and l_2, $l_1 = l_2$ means that $\mathcal{I} \Vdash l_1$ if and only if $\mathcal{I} \Vdash l_2$ for any interpretation $\mathcal{I}$. The algebraic laws given in Table 6 are useful. We prove only some of them.

Proof of R5

$(\sigma, k, j) \Vdash l \cdot l^*$
$\iff$ there exists r, $k \leq r \preceq j$, such that $(\sigma, k, r) \Vdash l$ and $(\sigma, r, j) \Vdash l^*$
$\iff$ there exists r, $k \leq r \preceq j$, such that $(\sigma, k, r) \Vdash l$, and $r = j$ or
there exist finitely many integers $r = r_0 \leq r_1 \leq \ldots \preceq r_n = j$ such that for all h,
$1 \leq h \leq n$, $(\sigma, r_{h-1}, r_h) \Vdash l$
$\iff$ $(\sigma, k, j) \Vdash l$ or
there exist finitely many integers $k = r'_0 \leq r = r_0 = r'_1 \leq r_1 = r'_2 \leq \ldots \preceq r_{n-1} = r'_n$
such that for all h, $1 \leq h \leq n$, $(\sigma, r'_{h-1}, r'_h) \Vdash l$, and $(\sigma, r'_n, j) \Vdash l$
$\iff$ $(\sigma, k, j) \Vdash l$ or
there exists r'_n, $k \leq r'_n \preceq j$, such that $(\sigma, k, r'_n) \Vdash l^*$ and $(\sigma, r'_n, j) \Vdash l$
$\iff$ $(\sigma, k, k) \Vdash l^*$ and $(\sigma, k, j) \Vdash l$ or
there exists r'_n, $k \leq r'_n \preceq j$, such that $(\sigma, k, r'_n) \Vdash l^*$ and $(\sigma, r'_n, j) \Vdash l$
$\iff$ there exists r', $k \leq r' \preceq j$, such that $(\sigma, k, r) \Vdash l^*$ and $(\sigma, r, j) \Vdash l$
$\iff$ $(\sigma, k, j) \Vdash l^* \cdot l$

Proof of R8

$(\sigma, k, j) \Vdash l^*$
$\iff$ $j = k$ or there exist finitely many integers $k = r_0 \leq r_1 \leq \ldots \preceq r_n = j$ such that
for all h, $1 \leq h \leq n$, $(\sigma, r_{h-1}, r_h) \Vdash l$
$\iff$ $(\sigma, k, j) \Vdash \epsilon$ or
there exists r_1, $k \leq r_1 \preceq j$ such that $(\sigma, k, r_1) \Vdash l$ and $(\sigma, r_1, j) \Vdash l^*$
$\iff$ $(\sigma, k, j) \Vdash \epsilon$ or $(\sigma, k, j) \Vdash l \cdot l^*$
$\iff$ $(\sigma, k, j) \Vdash \epsilon \otimes (l \cdot l^*)$

Proof of R13

$(\sigma, k, j) \Vdash l_1 \cdot (l_2 \otimes l_3)$
$\iff$ there exists r, $k \leq r \preceq j$, such that $(\sigma, k, r) \Vdash l_1$ and $(\sigma, r, j) \Vdash l_2 \otimes l_3$
$\iff$ there exists r, $k \leq r \preceq j$, such that $(\sigma, k, r) \Vdash l_1$ and $(\sigma, r, j) \Vdash l_2$, or
$(\sigma, k, r) \Vdash l_1$ and $(\sigma, r, j) \Vdash l_3$
$\iff$ $(\sigma, k, j) \Vdash l_1 \cdot l_2$ or $(\sigma, k, j) \Vdash l_1 \cdot l_3$
$\iff$ $(\sigma, k, j) \Vdash (l_1 \cdot l_2) \otimes (l_1 \cdot l_3)$

PROOF OF R15

$$
\begin{aligned}
& (\sigma, k, j) \Vdash l_1 \cdot (l_2 \cdot l_1)^* \\
\Longleftrightarrow\ & \text{there exists } r,\ k \le r \preceq j, \text{ such that } (\sigma, k, r) \Vdash l_1 \text{ and } (\sigma, r, j) \Vdash (l_2 \cdot l_1)^* \\
\Longleftrightarrow\ & \text{there exists } r,\ k \le r \preceq j, \text{ such that } (\sigma, k, r) \Vdash l_1 \text{ and } r = j, \text{ or} \\
& \text{there exists } r,\ k \le r \preceq j, \text{ such that } (\sigma, k, r) \Vdash l_1 \text{ and there exist finitely many} \\
& \text{integers } r = r_0 \le r_1 \le \ldots \preceq r_n = j, \text{ such that for all } h,\ 1 \le h \le n,\ (\sigma, r_{h-1}, r_h) \Vdash l_2 \cdot l_1 \\
\Longleftrightarrow\ & \text{there exists } r,\ k \le r \preceq j, \text{ such that } (\sigma, k, r) \Vdash l_1 \text{ and } r = j, \text{ or} \\
& \text{there exists } r,\ k \le r \preceq j, \text{ such that } (\sigma, k, r) \Vdash l_1 \text{ and there exist finitely many} \\
& \text{integers } r = r_0 \le r_1 \le \ldots \preceq r_n = j, \text{ such that for all } h,\ 1 \le h \le n, \text{ there exists } t_h, \\
& r_{h-1} \le t_h \le r_h, \text{ such that } (\sigma, r_{h-1}, t_h) \Vdash l_2 \text{ and } (\sigma, t_h, r_h) \Vdash l_1 \\
\Longleftrightarrow\ & (\sigma, k, k) \Vdash \epsilon \text{ and } (\sigma, k, j) \Vdash l_1, \text{ or} \\
& \text{there exist finitely many integers } k = t_0 \le t_1 \le \ldots \preceq t_n, \text{ such that for all } h, \\
& 1 \le h \le n,\ (\sigma, t_{h-1}, t_h) \Vdash l_1 \cdot l_2, \text{ and } (\sigma, t_n, j) \Vdash l_1 \\
\Longleftrightarrow\ & (\sigma, k, j) \Vdash (l_1 \cdot l_2)^* \cdot l_1
\end{aligned}
$$

2.2.2. CCM

Syntax The syntax of CCM is given as follows:

$$CCM ::= P \circ\!\!\vee (l) \mid CCM_1 \parallel CCM_2$$

where P is a PPTL formula and l a sequence expression. Over ($\circ\!\!\vee$) and parallel ($\parallel$) are temporal operators. Hence, all of the CCM formulas are temporal formulas.

Semantics For a CCM formulas $P \circ\!\!\vee (l)$, if it is satisfiable, the interpretation of P is controlled by the sequence expressions l. The beginning and ending points of the interpretation of each natural number n in l constitute the coarse-grained interval of P. Therefore, to give the semantics of CCM formulas, it is necessary to define the set of endpoint lists denoted by $S_l^{\mathcal{I}}$.

Definition 1 *(Concatenation of two strings).* Given two strings $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$, the concatenation of X and Y is defined by:

$$X \cdot Y = (x_1, \ldots, x_m) \cdot (y_1, \ldots, y_n) = (x_1, \ldots, x_m, y_1, \ldots, y_n)$$

Definition 2 *(Concatenation of two sets of strings).* Given two sets S_1 and S_2, of strings, the concatenation of S_1 and S_2 is defined by:

$$S_1 \cdot S_2 = \{X \cdot Y \mid X \in S_1 \text{ and } Y \in S_2\}$$

Definition 3 *(Set of endpoint lists $S_l^{\mathcal{I}}$).* Let $\mathcal{I}$ be an arbitrary interpretation (σ, k, j). $S_l^{\mathcal{I}}$ is inductively defined as follows:

1. $S_{\emptyset}^{\mathcal{I}} = \emptyset$.
2. If $\mathcal{I} \Vdash \epsilon$, then $S_{\epsilon}^{\mathcal{I}} = \{(k, j)\}$.
3. If $\mathcal{I} \Vdash n$, then $S_n^{\mathcal{I}} = \{(k, j)\}$.
4. If $\mathcal{I} \Vdash l_1 \cdot l_2$, then $S_{l_1 \cdot l_2}^{\mathcal{I}} = \left\{ \tau \;\middle|\; \begin{array}{l} \text{there exists } r, k \le r \preceq j, \text{ such that} \\ \mathcal{I}_1 = (\sigma, k, r) \Vdash l_1 \text{ and } \mathcal{I}_2 = (\sigma, r, j) \Vdash l_2 \text{ and} \\ \tau \in S_{l_1}^{\mathcal{I}_1} \cdot S_{l_2}^{\mathcal{I}_2} \end{array} \right\}$
5. If $\mathcal{I} \Vdash l_1 \otimes l_2$, then $S_{l_1 \otimes l_2}^{\mathcal{I}} = S_{l_1}^{\mathcal{I}} \bigcup S_{l_2}^{\mathcal{I}}$.
6. If $\mathcal{I} \Vdash l^*$, then $S_{l^*}^{\mathcal{I}} = S_{\epsilon}^{\mathcal{I}} \cup \left\{ \tau \;\middle|\; \begin{array}{l} \text{there exist finitely many integers } k = r_0 \le r_1 \ldots \preceq r_n = j \\ \text{such that for all } 1 \le h \le n, \mathcal{I}_h = (\sigma, r_{h-1}, r_h) \Vdash l \text{ and} \\ \tau \in S_l^{\mathcal{I}_1} \cdot S_l^{\mathcal{I}_2} \cdot \ldots \cdot S_l^{\mathcal{I}_n} \end{array} \right\}$

From Definition 3, we have if $\mathcal{I} \nVdash l$, then $S_l^{\mathcal{I}} = \emptyset$. For example, the derivation process of $S_{(3 \otimes 5) \cdot 2^*}^{(\sigma, 0, 9)}$ is given as follows:

$$
\begin{aligned}
& (\sigma, 0, 9) \Vdash (3 \otimes 5) \cdot 2^* \\
\Longleftrightarrow\ & (\sigma, 0, 9) \Vdash (3 \cdot 2^*) \otimes (5 \cdot 2^*) \\
\Longleftrightarrow\ & (\sigma, 0, 9) \Vdash 3 \cdot 2^* \text{ or } (\sigma, 0, 9) \Vdash 5 \cdot 2^* \\
\Longleftrightarrow\ & (\sigma, 0, 3) \Vdash 3 \text{ and } (\sigma, 3, 9) \Vdash 2^*, \text{ or} \\
& (\sigma, 0, 5) \Vdash 5 \text{ and } (\sigma, 5, 9) \Vdash 2^*
\end{aligned}
$$

Table 7
Semantics of cylinder computation model.

1. $\mathcal{I} \models P \circ\!\!\circ (l)$ iff one of the following cases holds:
   (a) $\mathcal{I} \Vdash l$ and there exists $(r_0, r_1, \ldots, r_n) \in S_l^{\mathcal{I}}$, $n \in N_0$ such that $\sigma' = \sigma \downarrow (r_0, r_1, \ldots, r_h)$ for some $0 \le h \le n$ and $(\sigma', 0, |\sigma'|) \models P$, or
   (b) there exists r, $k \le r \preceq j$ such that $\mathcal{I}_1 = (\sigma, k, r) \Vdash l$ and there exists $(r_0, r_1, \ldots, r_n) \in S_l^{\mathcal{I}_1}$, $n \in N_0$ and $\sigma' = \sigma \downarrow (r_0, r_1, \ldots, r_n) \bullet \sigma_{(r_n+1..j)}$ and $(\sigma', 0, |\sigma'|) \models P$.
2. $\mathcal{I} \models CCM_1 \parallel CCM_2$ iff one of the following cases holds:
   (a) $\mathcal{I} \models CCM_1$ and $(\sigma, k, r) \models CCM_2$; *true*, or
   (b) $\mathcal{I} \models CCM_2$ and $(\sigma, k, r) \models CCM_1$; *true*.

$$
\begin{aligned}
&\Longleftrightarrow (\sigma,0,3) \Vdash 3 \text{ and } (\sigma,3,5) \Vdash 2 \text{ and } (\sigma,5,7) \Vdash 2 \text{ and } (\sigma,7,9) \Vdash 2, \text{ or}\\
&\quad\ (\sigma,0,5) \Vdash 5 \text{ and } (\sigma,5,7) \Vdash 2 \text{ and } (\sigma,7,9) \Vdash 2\\
&\Longrightarrow S_3^{(\sigma,0,3)} = \{(0,3)\} \text{ and } S_2^{(\sigma,3,5)} = \{(3,5)\} \text{ and } S_2^{(\sigma,5,7)} = \{(5,7)\} \text{ and } S_2^{(\sigma,7,9)} = \{(7,9)\}\\
&\quad\ S_5^{(\sigma,0,5)} = \{(0,5)\} \text{ and } S_2^{(\sigma,5,7)} = \{(5,7)\} \text{ and } S_2^{(\sigma,7,9)} = \{(7,9)\}\\
&\Longrightarrow S_3^{(\sigma,0,3)} = \{(0,3)\} \text{ and } S_{2^*}^{(\sigma,3,9)} = \{(3,5,5,7,7,9)\}\\
&\quad\ S_5^{(\sigma,0,5)} = \{(0,5)\} \text{ and } S_{2^*}^{(\sigma,5,9)} = \{(5,7,7,9)\}\\
&\Longrightarrow S_{3\cdot 2^*}^{(\sigma,0,9)} = \{(0,3,3,5,5,7,7,9)\}, \quad S_{5\cdot 2^*}^{(\sigma,0,9)} = \{(0,5,5,7,7,9)\}\\
&\Longrightarrow S_{(3\otimes 5)\cdot 2^*}^{(\sigma,0,9)} = S_{3\cdot 2^*}^{(\sigma,0,9)} \cup S_{5\cdot 2^*}^{(\sigma,0,9)} = \{(0,3,3,5,5,7,7,9), \quad (0,5,5,7,7,9)\}
\end{aligned}
$$

The semantics of Cylinder Computation Model is also defined by a satisfaction relation $\models$ by means of the interpretation $\mathcal{I} = (\sigma, k, j)$, given in Table 7.

In fact, each element in $S_l^{\mathcal{I}}$ is a sequence of non-negative integers, and a sequence expression can be satisfied by an interpretation in more than one way. Each element in $S_l^{\mathcal{I}}$ records a particular way in which $\mathcal{I}$ satisfies l, containing all of the endpoints. The definition of $S_l^{\mathcal{I}}$ is useful since a PPTL formula P in CCM formulas is interpreted over a coarse-grained interval composed of the points from one of the elements in $S_l^{\mathcal{I}}$.

3. Propositional projection temporal logic with cylinder computation model

To model, specify and verify multi-core parallel programs, Propositional Projection Temporal Logic with Cylinder Computation Model (CCM–PPTL) is proposed, which is an extension of PPTL by including CCM.

3.1. Syntax

The formulas of CCM–PPTL are defined as follows:

$$
\begin{aligned}
\beta ::= {} & p \mid \neg\beta \mid \bigcirc \beta \mid \beta_1 \vee \beta_2 \mid (\beta_1, \ldots, \beta_m)\ prj\ \beta\\
& \mid (\beta_1, \ldots, (\beta_u, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta \mid CCM
\end{aligned}
$$

where $p \in Prop$ is an arbitrary atomic proposition; β, $\beta_1, \ldots$ and β_m are arbitrary CCM–PPTL formulas; CCM is an arbitrary CCM formula defined in Section 2.2.2.

3.2. Semantics

The semantics of CCM–PPTL is also defined as a satisfaction relation $\models$ by means of the interpretation $\mathcal{I} = (\sigma, k, j)$, given in Table 8.

Note that the formula P appearing in $P \circ\!\!\circ (l)$ is a PPTL formula, not a CCM–PPTL formula. That is why the syntax and semantics of PPTL and CCM–PPTL have to be defined separately. CCM is of a typical form of $P_1 \circ\!\!\circ (l_1) \parallel \cdots \parallel P_m \circ\!\!\circ (l_m)$ where each P_i is a PPTL formula while each l_i is a sequence expression. With this parallelism, a main time interval is the sequence of fine-grained unit subintervals with length one while several coarse-grained projected intervals over which processes are interpreted are in parallel with the main time interval. This computation model can be viewed as m processes that share one processor and each occupies an execution core cooperating to complete their tasks in a parallel way. Each process progresses in its own speed and communicates with each other at some global states. This computation model realizes the coordination among all processes. Sequence expression l_i is used to control and determine the execution points (states) of P_i. Parallel ($\parallel$) is the main operator in CCM. Thus $P_1 \circ\!\!\circ (l_1) \parallel \cdots \parallel P_m \circ\!\!\circ (l_m)$ is endowed with the semantics of multi-core parallel computing. For example, the interval satisfying CCM formula $P_1 \circ\!\!\circ (2\cdot 1\cdot 2\cdot 3\cdot 3\cdot 1\cdot 5) \parallel P_2 \circ\!\!\circ (2\cdot 3\cdot 3\cdot 4) \parallel P_3 \circ\!\!\circ (3\cdot 2\cdot 5\cdot 7) \parallel P_4 \circ\!\!\circ ((1\cdot 4)^*) \parallel P_5 \circ\!\!\circ (2^*)$ is given in Fig. 1.

Table 8
Semantics of CCM.

1. $\mathcal{I} \models p$ iff $s_k[p] = true$ for any atomic proposition p.
2. $\mathcal{I} \models \neg\beta$ iff $\mathcal{I} \not\models \beta$.
3. $\mathcal{I} \models \bigcirc\beta$ iff $(\sigma, k+1, j) \models \beta$.
4. $\mathcal{I} \models \beta_1 \vee \beta_2$ iff $\mathcal{I} \models \beta_1$ or $\mathcal{I} \models \beta_2$.
5. $\mathcal{I} \models (\beta_1, \ldots, \beta_m)$ *prj* β iff
   there exist $k = r_0 \leq r_1 \leq \ldots \leq r_{m-1} \leq r_m \leq j$ such that for all $1 \leq l \leq m$, $(\sigma, r_{l-1}, r_l) \models \beta_l$, and $(\sigma', 0, |\sigma'|) \models \beta$ for one of the following σ':
   (a) $r_m < j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_m) \bullet \sigma_{(r_m+1..j)}$;
   (b) $r_m = j$ and $\sigma' = \sigma \downarrow (r_0, \ldots, r_h)$ for some $0 \leq h \leq m$.
6. $\mathcal{I} \models (\beta_1, \ldots, (\beta_u, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)$ *prj* β iff one of the following cases holds:
   (a) $\mathcal{I} \models (\beta_1, \ldots, (\beta_u, \ldots, \beta_l)^{(n)}, \ldots, \beta_m)$ *prj* β for some $n \geq 1$ and $n \in N_0$;
   (b) $l = m$ and $j = \omega$ and there exist infinitely many integers $k = r_0 \leq r_1 \leq \cdots$ and $\lim_{x\to\infty} r_x = \omega$, such that
   - $(\sigma, r_{x-1}, r_x) \models \beta_x$ for $1 \leq x \leq u-1$, and
   - $(\sigma, r_{u+t(l-u+1)+n-1}, r_{u+t(l-u+1)+n}) \models \beta_{u+n}$, for $t \geq 0$ and $0 \leq n \leq l-u$, and
   - $(\sigma', 0, |\sigma'|) \models \beta$ and $\sigma' = \sigma \downarrow (r_0, r_1, \ldots, r_h)$ for some $h \in N_\omega$.
7. $\mathcal{I} \models CCM$ see the semantics of CCM.

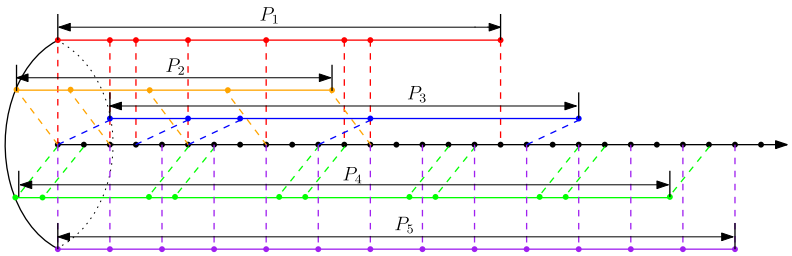

Fig. 1. P_1 ov $(2 \cdot 1 \cdot 2 \cdot 3 \cdot 3 \cdot 1 \cdot 5) \parallel P_2$ ov $(2 \cdot 3 \cdot 3 \cdot 4) \parallel P_3$ ov $(3 \cdot 2 \cdot 5 \cdot 7) \parallel P_4$ ov $((1 \cdot 4)^*) \parallel P_5$ ov (2^*).

Table 9
Logical Laws of CCM.

L1	P ov $(0) \equiv P$
L2	P ov $(\emptyset) \equiv false$
L3	ε ov $(n) \equiv \bigcirc(\varepsilon$ ov $(n-1))$
L4	ε ov $(n \cdot l) \equiv \bigcirc(\varepsilon$ ov $(n-1 \cdot l))$
L5	$\bigcirc P$ ov $(n) \equiv \bigcirc^n \varepsilon; P$
L6	$\bigcirc P$ ov $(n \cdot l) \equiv \bigcirc^n \varepsilon; (P$ ov $(l))$
L7	$(w \wedge P)$ ov $(l) \equiv w \wedge (P$ ov $(l))$
L8	P ov $(l_1 \otimes l_2) \equiv (P$ ov $(l_1)) \vee (P$ ov $(l_2))$
L9	$(P_1 \vee P_2)$ ov $(l) \equiv (P_1$ ov $(l)) \vee (P_2$ ov $(l))$
L10	ε ov $(n^*) \cong \varepsilon \vee (\mathsf{len}(n)^*; len(n))$

3.2.1. Logical laws

All the logical laws in PPTL also hold in CCM–PPTL. For more details, refer to [6]. In Table 9, we present the logical laws regarding the " ov " operator in CCM formulas. We will use them to transform all the CCM formulas into their normal forms.

3.2.2. Some definitions

We now define a normal form and a complete normal form for CCM–PPTL formulas upon which the completeness of the axiom system given later is proved.

Definition 4 *(Normal form).* Let β_p be the set of atomic propositions appearing in a CCM–PPTL formula β. The normal form of β can be defined as follows.

$$\beta \equiv \bigvee_{j=0}^{n_0} (\beta_{ej} \wedge \varepsilon) \vee \bigvee_{i=0}^{n_1} (\beta_{ci} \wedge \bigcirc \beta_i')$$

where $\beta_{ej} \equiv \bigwedge_{k=1}^{m_j} \dot{p}_{jk}$, $\beta_{ci} \equiv \bigwedge_{h=1}^{t_i} \dot{q}_{ih}$, $p_{jk}, q_{ih} \in \beta_p$, for any $r \in \beta_p$, $\dot{r}$ denotes r or $\neg r$; β_i' is a CCM–PPTL formula without "$\vee$" being the main operator.

Table 10
Transformation rules on sequence expressions.

S1	$\epsilon \simeq \epsilon^* \simeq 0 \simeq 0^*$
S2	$0 \cdot l \simeq l \cdot 0 \simeq l$
S3	$l_1 \cdot (l_2 \cdot l_3) \simeq (l_1 \cdot l_2) \cdot l_3 \simeq l_1 \cdot l_2 \cdot l_3$
S4	$l_1 \cdot (l_2 \otimes l_3) \cdot l_4 \simeq (l_1 \cdot l_2 \cdot l_4) \otimes (l_1 \cdot l_3 \cdot l_4)$
S5	$l^* \simeq \epsilon \otimes (l \cdot l^*) \simeq (\epsilon \otimes l)^*$
S6	$l \cdot l^* \simeq l^* \cdot l$
S7	$l^* \cdot l^* \simeq l^*$
S8	$(l^*)^* \simeq l^*$
S9	$l_1 \cdot (l_2 \cdot l_1)^* \simeq (l_1 \cdot l_2)^* \cdot l_1$
S10	$(0 \otimes l)^* \simeq l^*$
S11	$l_1 \simeq l_2 \Longrightarrow l \simeq l[l_2/l_1]$
S12	$l \simeq (l_1 \cdot l) \otimes l_2 \Longrightarrow l \simeq l_1^* \cdot l_2$

Table 11
Axioms on CCM.

A1	$P \circledcirc (l_1 \cdot \emptyset \cdot l_2) \cong \mathit{false}$
A2	$P \circledcirc (0) \cong P$
A3	$\varepsilon \circledcirc (m \cdot l) \cong \bigcirc(\varepsilon \circledcirc (m-1 \cdot l)) \quad (m > 0)$
A4	$\bigcirc P \circledcirc (m \cdot l) \cong \bigcirc^m \varepsilon; (P \circledcirc (l)) \quad (m > 0)$
A5	$(w \wedge P) \circledcirc (l) \cong w \wedge (P \circledcirc (l))$
A6	$P \circledcirc (l_1 \otimes l_2) \cong (P \circledcirc (l_1)) \vee (P \circledcirc (l_2))$
A7	$(P_1 \vee P_2) \circledcirc (l) \supset (P_1 \circledcirc (l)) \vee (P_2 \circledcirc (l))$
A8	$\varepsilon \circledcirc (n^*) \cong \varepsilon \vee (\mathsf{len}(n)^*; \mathit{len}(n))$
A9	$CCM_1 \parallel CCM_2 \cong (CCM_1; \mathit{true}) \wedge CCM_2 \vee (CCM_2; \mathit{true}) \wedge CCM_1$

Table 12
Inference rules on CCM.

I1	$P \supset P' \Longrightarrow P \circledcirc (l) \supset P' \circledcirc (l)$
I2	$l_1 \simeq l_2 \Longrightarrow P \circledcirc (l_1) \cong P \circledcirc (l_2)$

Definition 5 *(Complete normal form).* Let β_p be the set of atomic propositions appearing in a CCM–PPTL formula β. The normal form of β can be defined as follows.

$$\beta \equiv \bigvee_{j=0}^{n_0} (\beta_{ej} \wedge \varepsilon) \vee \bigvee_{i=0}^{n_1} (\beta_{ci} \wedge \bigcirc \beta_i')$$

where $\beta_{ej} \equiv \bigwedge_{k=1}^{m_j} \dot{p}_{jk}$, $\beta_{ci} \equiv \bigwedge_{h=1}^{t_i} \dot{q}_{ih}$, $p_{jk}, q_{ih} \in \beta_p$, for any $r \in \beta_p$, $\dot{r}$ denotes r or $\neg r$; further $\bigvee_i \beta_{ci} \equiv \mathsf{true}$ and $\bigvee_{i \neq j} (\beta_{ci} \wedge \beta_{cj}) \equiv \mathsf{false}$; β_i' is an arbitrary CCM–PPTL formula.

Note that a complete normal form may be not a normal form, since "$\vee$" is possible the main operator of β_i'.

Definition 6 *(Terminable formula and non-terminable formula).* For any CCM–PPTL formula β, if $\beta \wedge \Diamond \varepsilon \not\equiv \mathit{false}$, then β is a terminable formula. Otherwise, it is a non-terminable formula.

4. Axiom system of CCM–PPTL

In this section, we give an axiom system for CCM–PPTL which is based on the axiom system of PPTL. Some axioms and inference rules on CCM structure are included.

4.1. Axiom system

4.1.1. Transformation rules on sequence expressions

Since the deduction of CCM formulas depends on the nature of sequence expressions, some essential transformation rules on sequence expressions (see Table 10) need to be included in the axiom system of CCM–PPTL.

4.1.2. Axioms and Inference Rules on CCM

The axioms and inference rules on CCM formulas, given in Tables 11 and 12, respectively, are based on the transformation rules of sequence expressions.

A1 means that any CCM formula $P \circledcirc (l)$ with an unsatisfiable sequence expression l is also unsatisfiable. A2 indicates that any CCM formula $P \circledcirc (l)$ with its sequence expression l being 0 is deduced to the PPTL formula P. A3 tells us that

if a sequence expression begins with a positive integer m and the PPTL formula is ε, then we can extract a next operator directly with m decreasing by 1. A4 shows that if a sequence expression begins with a positive integer m and the PPTL formula is a next formula $\bigcirc P$, then we can extract m next operators directly with deleting m from the sequence expression and the next operator from $\bigcirc P$. A5 means that if the PPTL formula contains a conjunct being a state formula w, then w can be extracted from the PPTL formula and treated as a conjunct of the whole formula. A6 indicates the distributivity of the sum operator $\otimes$ over the $\circ\vee$ operator. A7 describes the distributivity of the disjunct connective $\vee$ over the $\circ\vee$ operator. A8 means if the PPTL formula is ε and the sequence expression is an iteration of a natural number n, then the CCM formula describe the intervals with length which can be divided by n. A9 presents the semantics of the parallel operator, that is, CCM_1 and CCM_2 are interpreted in parallel and can specify their own lengths of expressions.

I1 indicates that the implication between two PPTL formulas is preserved by the CCM structure. I2 means that if l_1 is equivalent to l_2, then two CCM formulas with l_1 and l_2 as sequence expressions respectively are also equivalent. In the following, the soundness and completeness of the axiom system of CCM–PPTL are proved.

4.2. Soundness

Theorem 1 *(Soundness). For any CCM–PPTL formula β, if $\vdash \beta$, then $\models \beta$.*

Proof. We need to prove each axiom in the proof system of CCM–PPTL is valid in the model theory of CCM–PPTL and each inference rule preserves the validity of premises. Since the proof system of PPTL is sound, we only need to consider the axioms and inference rules on the CCM structure. After a careful observation, we find that each transformation rule of sequence expressions is an algebraic law and each axiom on CCM is also a logical law. Two inference rules, I1 and I2, which formalize the idea of substitution, are easy to be proved. All the above ensure the soundness of the proof system of CCM–PPTL. □

4.3. Completeness

To prove the completeness of the axiom system given in Theorem 5, seven lemmas are proved in advance. In general, the set of CCM–PPTL formulas are partitioned into two categories: terminable and non-terminable formulas. We will prove that a terminable formula is satisfiable (Lemma 5), and that for a non-terminable formula β satisfying $\nvdash \beta \rightarrow \mathit{false}$, β is also satisfiable (Lemma 7). Lemma 7 is based on a fact that any CCM–PPTL formula can be deduced into a normal form in the axiom system of CCM–PPTL, which is given in Theorem 2.

Lemma 1. *For a CCM–PPTL formula β, if $\beta \cong \beta'$ where β' is in normal form, there exists a CCM–PPTL formula β_c in complete normal form satisfying $\beta \cong \beta_c$.*

Proof. In this proof, it is essential to give a construction method for complete normal form from a normal form of β. We conform to the method given in [6,9]. Suppose that one of the normal forms of β is $\alpha_e \wedge \varepsilon \vee \bigvee_{i=1}^{n} \alpha_i \wedge \bigcirc \beta_i$ where α_e and the α_i's are state formulas. Following the standard definitions of min-term, max-term and basic sum in classical propositional logic, we treat each state formula α_i as an atomic proposition and construct 2^n min-terms, $m_0, \ldots, m_{2^n-1}$. Moreover, we treat β_i as an atomic proposition and construct 2^n max-terms, $M_0, \ldots, M_{2^n-1}$. From these max-terms, we obtain 2^n basic sums, $M'_0, \ldots, M'_{2^n-1}$, where M'_j denotes M_{2^n-1-j} after deleting negative disjunctions and M'_0 denotes *false*. It is easy to deduce the following theorem just using the axioms of the classical propositional logic in CCM–PPTL proof system. Hence, the proof is omitted here.

$$\beta \cong \alpha_e \wedge \varepsilon \vee \bigvee_{i=1}^{n} (\alpha_i \wedge \bigcirc \beta_i) \cong \alpha_e \wedge \varepsilon \vee \bigvee_{i=0}^{2^n-1} (m_i \wedge \bigcirc M'_i) \qquad \square$$

Lemma 2. *Let $\alpha_1, \ldots, \alpha_n$ be state formulas, and β_i a general CCM–PPTL formula. If $\bigvee_{i=1}^{n} \alpha_i \cong \mathit{true}$ and $\bigvee_{i \neq j} \alpha_i \wedge \alpha_j \cong \mathit{false}$, then $\neg(\bigvee_{i=1}^{n} \alpha_i \wedge \beta_i) \cong \bigvee_{i=1}^{n} (\alpha_i \wedge \neg \beta_i)$.*

Proof. It is a proof in the classical propositional logic. Hence, it is omitted here. □

Lemma 1 indicates that any normal form can be deduced into a complete normal form. In the deduction of $\neg\beta$ into a normal form as we will see later on in Theorem 5, β is deduced into its normal form first, then further deduced into its complete normal form using Lemma 1. Finally, we deduce $\neg\beta$ into its normal form using Lemma 2 based on β's complete normal form. To deduce the CCM construct into a normal form, first we need to deduce the sequence expression into one of three forms using the transformation rules, which is formalized in the following lemma.

Lemma 3. *For any sequence expression l, l can be deduced using the transformation rules into one of the following forms:*
(Form 1) $\emptyset$
(Form 2) $0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i)$ where $m \geq 0$ and $n_i \in N$ for all $1 \leq i \leq m$.
(Form 3) $\bigotimes_{i=1}^{m}(n_i \cdot l_i)$ where $m \geq 1$ and $n_i \in N$ for all $1 \leq i \leq m$.

Proof. The proof proceeds by induction on the structure of sequence expressions.
Base:

(1) l is $\emptyset$, then it is already in Form 1.
(2) l is ϵ, then $l \simeq 0$ according to the algebraic laws of sequence expressions, which is in Form 2 under the condition $m = 0$.
(3) l is n, if n is zero, it is the same as case (2); if n is a positive integer, according to the algebraic laws we have $n \simeq (n \cdot 0)$, which is in Form 3 under the condition $m = 1$, $n_1 = n$ and $l_1 = 0$.

Induction:

(4) l is $(l_1 \cdot l_2)$, with the hypothesis that both of l_1 and l_2 can be transformed into one of the three forms, then there are 3×3 possible combinations.

If l_1 or l_2 is transformed into $\emptyset$, which covers 5 possible combinations, l can be equivalently transformed into $\emptyset$ which is in Form 1.

If both of the transformations of l_1 and l_2 are in Form 2, we have the following computation

$$\begin{aligned}
l &\simeq (l_1 \cdot l_2) \\
&\simeq (0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i)) \cdot (0 \otimes \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)) && \text{Hypothesis} \\
&\simeq (0 \cdot (0 \otimes \bigotimes_{j=1}^{n}(n'_j \cdot l'_j))) \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot (0 \otimes \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)))) && \text{S4} \\
&\simeq (0 \cdot 0) \otimes (0 \cdot \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)) \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot 0) \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)) && \text{S4} \\
&\simeq 0 \otimes \bigotimes_{j=1}^{n}(n'_j \cdot l'_j) \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i) \otimes \bigotimes_{i=1}^{m}\bigotimes_{j=1}^{n}(n_i \cdot l_i \cdot n'_j \cdot l'_j) && \text{S2, S4}
\end{aligned}$$

Hence, l is transformed into Form 2.

If l_1 is in Form 2 and l_2 Form 3, then

$$\begin{aligned}
l &\simeq (l_1 \cdot l_2) \\
&\simeq (0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i)) \cdot \bigotimes_{j=1}^{n}(n'_j \cdot l'_j) && \text{Hypothesis} \\
&\simeq (0 \cdot \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)) \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot \bigotimes_{j=1}^{n}(n'_j \cdot l'_j)) && \text{S4} \\
&\simeq \bigotimes_{j=1}^{n}(n'_j \cdot l'_j) \otimes \bigotimes_{i=1}^{m}\bigotimes_{j=1}^{n}(n_i \cdot l_i \cdot n'_j \cdot l'_j) && \text{S2, S4}
\end{aligned}$$

Hence, l is in Form 3.

If l_1 is in Form 3 and l_2 Form 2, l will be transformed into Form 3. If both l_1 and l_2 are in Form 3, l will be in Form 3. The proofs with these two cases are similar as the proof given above, hence they are omitted here.

(5) l is $l_1 \otimes l_2$, with the hypothesis that l_1 and l_2 are transformed into l'_1 and l'_2, then $l'_1 \otimes l'_2$ is already in one of the three forms.
(6) l is $(l')^*$, using the transformation rule S5, we have $l \simeq \epsilon \otimes (l' \cdot (l')^*)$; then using S1, we have $l \simeq 0 \otimes (l' \cdot (l')^*)$. Suppose that l' has been transformed into l''. If $l'' \simeq \emptyset$, then $l = 0$, which is in Form 2. If $l'' \simeq 0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i)$, then we have

$$\begin{aligned}
l &\simeq (l')^* \\
&\simeq (0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i))^* && \text{Hypothesis} \\
&\simeq (\bigotimes_{i=1}^{m}(n_i \cdot l_i))^* && \text{S10} \\
&\simeq 0 \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot (\bigotimes_{i=1}^{m}(n_i \cdot l_i))^*) && \text{S5} \\
&\simeq 0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i \cdot (\bigotimes_{j=1}^{m}(n_j \cdot l_j))^*) && \text{S4}
\end{aligned}$$

Hence l is in Form 2.

If l'' is in Form 3, that is, $l'' \simeq \bigotimes_{i=1}^{m}(n_i \cdot l_i)$, then we have

$$\begin{aligned}
l &\simeq (l')^* \\
&\simeq 0 \otimes (l' \cdot (l')^*) && \text{S5, S1} \\
&\simeq 0 \otimes ((\bigotimes_{i=1}^{m}(n_i \cdot l_i) \cdot (l')^*) && \text{Hypothesis} \\
&\simeq 0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i \cdot (l')^*) && \text{S4}
\end{aligned}$$

Hence l is in Form 2. □

Lemma 4. *For a CCM formula β, there exists a formula β' in normal form such that $\beta \cong \beta'$.*

Proof. The proof proceeds by induction on the syntax of CCM.
Base: For $P \circ\!\!\vee (l)$, by Lemma 3, if $l \simeq \emptyset$, then $P \circ\!\!\vee (l) \cong \mathit{false}$ (A1).
If $l \simeq 0 \otimes \bigotimes_{i=1}^{m}(n_i \cdot l_i)$, by the theorem that any PPTL formula P can be deduced into its normal form, we have

$$
\begin{aligned}
P \circ\!\!\vee (l) &\cong P \circ\!\!\vee (0 \otimes \textstyle\bigotimes_{i=1}^{m}(n_i \cdot l_i)) && \text{I2}\\
&\cong P \circ\!\!\vee (0) \vee \textstyle\bigvee_{i=1}^{m} P \circ\!\!\vee (n_i \cdot l_i) && \text{A6}\\
&\cong P \vee \textstyle\bigvee_{i=1}^{m} P \circ\!\!\vee (n_i \cdot l_i) && \text{A2}\\
&\cong P_e \wedge \varepsilon \vee \textstyle\bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj})\\
&\quad \vee \textstyle\bigvee_{i=1}^{m}(P_e \wedge \varepsilon \vee \bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj})) \circ\!\!\vee (n_i \cdot l_i) && \text{Hypothesis}\\
&\cong P_e \wedge \varepsilon \vee \textstyle\bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj})\\
&\quad \vee \textstyle\bigvee_{i=1}^{m}(P_e \wedge \varepsilon) \circ\!\!\vee (n_i \cdot l_i)\\
&\quad \vee \textstyle\bigvee_{i=1}^{m} \bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj}) \circ\!\!\vee (n_i \cdot l_i) && \text{A7}\\
&\cong P_e \wedge \varepsilon \vee \textstyle\bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj})\\
&\quad \vee \textstyle\bigvee_{i=1}^{m} P_e \wedge (\varepsilon \circ\!\!\vee (n_i \cdot l_i))\\
&\quad \vee \textstyle\bigvee_{i=1}^{m} \bigvee_{j=1}^{n} P_{cj} \wedge (\bigcirc P'_{cj} \circ\!\!\vee (n_i \cdot l_i)) && \text{A5}\\
&\cong P_e \wedge \varepsilon \vee \textstyle\bigvee_{j=1}^{n}(P_{cj} \wedge \bigcirc P'_{cj})\\
&\quad \vee \textstyle\bigvee_{i=1}^{m} P_e \wedge \bigcirc(\varepsilon \circ\!\!\vee (n_i - 1 \cdot l_i)) && \text{A3}\\
&\quad \vee \textstyle\bigvee_{i=1}^{m} \bigvee_{j=1}^{n} P_{cj} \wedge \bigcirc^{n_i}(P'_{cj} \circ\!\!\vee (l_i)) && \text{A4}
\end{aligned}
$$

If $l \simeq \bigotimes_{i=1}^{m}(n_i \cdot l_i)$, $P \circ\!\!\vee (l)$ can also be deduced into its normal form in a similar way.
Induction: For $CCM_1 \parallel CCM_2$, suppose that CCM_1 and CCM_2 have been deduced into their normals, then

$$
\begin{aligned}
& CCM_1 \parallel CCM_2\\
\cong\ & (CCM_1; \mathit{true}) \wedge CCM_2 \vee (CCM_2; \mathit{true}) \wedge CCM_1 && \text{A8}\\
\cong\ & ((\alpha_{1e} \wedge \varepsilon \vee \textstyle\bigvee_{i=1}^{m} \alpha_{1i} \wedge \bigcirc\beta_{1i}); \mathit{true}) \wedge (\alpha_{2e} \wedge \varepsilon \vee \bigvee_{j=1}^{n} \alpha_{2j} \wedge \bigcirc\beta_{2j})\\
& \vee((\alpha_{2e} \wedge \varepsilon \vee \textstyle\bigvee_{j=1}^{n} \alpha_{2j} \wedge \bigcirc\beta_{2j}); \mathit{true}) \wedge (\alpha_{1e} \wedge \varepsilon \vee \bigvee_{i=1}^{m} \alpha_{1i} \wedge \bigcirc\beta_{1i}) && \text{Hypothesis}\\
\cong\ & ((\alpha_{1e} \wedge \varepsilon; \mathit{true}) \vee \textstyle\bigvee_{i=1}^{m}(\alpha_{1i} \wedge \bigcirc\beta_{1i}; \mathit{true})) \wedge (\alpha_{2e} \wedge \varepsilon \vee \bigvee_{j=1}^{n} \alpha_{2j} \wedge \bigcirc\beta_{2j})\\
& \vee((\alpha_{2e} \wedge \varepsilon; \mathit{true}) \vee \textstyle\bigvee_{j=1}^{n}(\alpha_{2j} \wedge \bigcirc\beta_{2j}; \mathit{true})) \wedge (\alpha_{1e} \wedge \varepsilon \vee \bigvee_{i=1}^{m} \alpha_{1i} \wedge \bigcirc\beta_{1i}) && \text{PDF, PEB}\\
\cong\ & (\alpha_{1e} \wedge \varepsilon; \mathit{true}) \wedge (\alpha_{2e} \wedge \varepsilon)\\
& \vee \textstyle\bigvee_{j=1}^{n}(\alpha_{1e} \wedge \varepsilon; \mathit{true}) \wedge (\alpha_{2j} \wedge \bigcirc\beta_{2j})\\
& \vee \textstyle\bigvee_{i=1}^{m} \bigvee_{j=1}^{n}(\alpha_{1i} \wedge \bigcirc\beta_{1i}; \mathit{true}) \wedge (\alpha_{2j} \wedge \bigcirc\beta_{2j})\\
& \vee \textstyle\bigvee_{i=1}(\alpha_{2e} \wedge \varepsilon; \mathit{true}) \wedge (\alpha_{1i} \wedge \bigcirc\beta_{1i})\\
& \vee \textstyle\bigvee_{j=1}^{n} \bigvee_{i=1}^{m}(\alpha_{2j} \wedge \bigcirc\beta_{2j}; \mathit{true}) \wedge (\alpha_{1i} \wedge \bigcirc\beta_{1i}) && \text{TAU}\\
\cong\ & \alpha_{1e} \wedge \alpha_{2e} \wedge \varepsilon\\
& \vee \textstyle\bigvee_{j=1}^{n} \alpha_{1e} \wedge \alpha_{2j} \wedge \bigcirc\beta_{2j}\\
& \vee \textstyle\bigvee_{i=1}^{m} \alpha_{2e} \wedge \alpha_{1i} \wedge \bigcirc\beta_{1i}\\
& \vee \textstyle\bigvee_{i=1}^{m} \bigvee_{j=1}^{n} \alpha_{1i} \wedge \alpha_{2j} \wedge \bigcirc((\beta_{1i}; \mathit{true}) \wedge \beta_{2j} \vee (\beta_{2j}; \mathit{true}) \wedge \beta_{1i}) && \text{PSM, PEB}
\end{aligned}
$$

□

Lemma 4 tells us that any CCM formula can be deduced into a normal form whenever its sequence expression has been deduced into one of three forms. This is also26009.9626200.9093104.496Tm()Tj47.1 31Tf9.9626009.9626298.39530T0 31Tf0Tc9.96260

Base:

(1) For any atomic proposition p, $p \cong p \wedge \varepsilon \vee p \wedge \bigcirc true$.
(2) For $\bigcirc\beta$, $\bigcirc\beta \cong true \wedge \bigcirc\beta$.

Induction:

(3) For $\neg\beta$, suppose that β can be deduced into its normal form, from Lemma 1, β can also be deduced into its complete normal form $\alpha_e \wedge \varepsilon \vee \vee_{i=1}^{r}(\alpha_i \wedge \bigcirc\beta_i)$ where $\vee_{i=1}^{r}\alpha_i \cong true$ and $\vee_{i\neq j}\alpha_i \wedge \alpha_j \cong false$. Then we have

$$\neg\beta \cong \neg(\alpha_e \wedge \varepsilon \vee \vee_{i=1}^{r}(\alpha_i \wedge \bigcirc\beta_i)) \cong \neg\alpha_e \wedge \varepsilon \vee \vee_{i=1}^{r}(\alpha_i \wedge \bigcirc\neg\beta_i)$$

(4) For $\beta_1 \vee \beta_2$, suppose β_1 and β_2 have been transformed into their normal form, then

$$\begin{aligned}\beta_1 \vee \beta_2 &\cong \alpha_{1e} \wedge \varepsilon \vee \vee_{i=1}^{m}\alpha_{1i} \wedge \bigcirc\beta_{1i} \vee \alpha_{2e} \wedge \varepsilon \vee \vee_{j=1}^{n}\alpha_{2j} \wedge \bigcirc\beta_{2j}\\ &\cong (\alpha_{1e} \vee \alpha_{2e}) \wedge \varepsilon \vee \vee_{i=1}^{m}\alpha_{1i} \wedge \bigcirc\beta_{1i} \vee \vee_{j=1}^{n}\alpha_{2j} \wedge \bigcirc\beta_{2j}\end{aligned}$$

(5) For $(\beta_1, \ldots, \beta_m)$ prj β_0, suppose that β_i' adheres to the following form:

$$\alpha_{ie} \wedge \varepsilon \vee \bigvee_{j=1}^{n_i}\alpha_{ij} \wedge \bigcirc\beta_{ij}$$

where α_{ie} and the α_{ij}'s are state formulas. The proof proceeds by induction on m.

Base: $m = 1$,

	β_1 prj β_0	
$\cong$	$(\alpha_{1e} \wedge \varepsilon \vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \bigcirc\beta_{1i})$ prj $(\alpha_{0e} \wedge \varepsilon \vee \bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j})$	Hypothesis
$\cong$	$\alpha_{1e} \wedge \varepsilon$ prj $\alpha_{0e} \wedge \varepsilon$	
	$\vee\alpha_{1e} \wedge \varepsilon$ prj $\bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j}$	
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \bigcirc\beta_{1i}$ prj $\alpha_{0e} \wedge \varepsilon$	
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \bigcirc\beta_{1i}$ prj $\bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j}$	PDF, PDB
$\cong$	$\alpha_{1e} \wedge \alpha_{0e} \wedge \varepsilon$	PEB, PEF, PSB, PSF
	$\vee \bigvee_{j=1}^{n_0}\alpha_{1e} \wedge \alpha_{0j} \wedge \bigcirc\beta_{0j}$	PDB, PSF, PSB, PEF
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \alpha_{0e} \wedge \bigcirc\beta_{1i}$	PDF, PSF, PSB, PEB
	$\vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}\alpha_{1i} \wedge \alpha_{0j} \wedge \bigcirc(\beta_{1i}; \beta_{0j})$	PSB, PSF, PDF, PDB

Induction: Suppose that $(\beta_2, \ldots, \beta_m)$ prj $\beta_0 \cong \alpha_e \wedge \varepsilon \vee \bigvee_{k=1}^{n}\alpha_k \wedge \bigcirc\beta_k'$. Then

	$(\beta_1, \beta_2, \ldots, \beta_m)$ prj β_0	
$\cong$	$(\alpha_{1e} \wedge \varepsilon, \beta_2, \ldots, \beta_m)$ prj β_0	Hypothesis
	$\vee \bigvee_{i=1}^{n_1}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, \beta_2, \ldots, \beta_m)$ prj β_0	PDF
$\cong$	$\alpha_{1e} \wedge (\beta_2, \ldots, \beta_m)$ prj β_0	PSF, PSM
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge (\bigcirc\beta_{1i}, \beta_2, \ldots, \beta_m)$ prj $(\alpha_{0e} \wedge \varepsilon)$	PSF, PDB
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge (\bigcirc\beta_{1i}, \beta_2, \ldots, \beta_m)$ prj $\bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j}$	PSF, PDB
$\cong$	$\alpha_{1e} \wedge \alpha_e \wedge \varepsilon \vee \bigvee_{k=1}^{n}\alpha_{1e} \wedge \alpha_k \wedge \bigcirc\beta_k'$	Hypothesis
	$\vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \alpha_{0e} \wedge \bigcirc(\beta_{1i}; \beta_2; \ldots; \beta_m)$	PSB, PNX
	$\vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}\alpha_{1i} \wedge \alpha_{0j} \wedge \bigcirc(\beta_{1i}; (\beta_2, \ldots, \beta_m)$ prj $\beta_{0j})$	PDB, PNX

(6) For $(\beta_1, \ldots, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)$ prj β_0, the proof proceeds in two steps.

In step one, we need to prove that if $\beta_i \cong \beta_i'$ $(0 \leq i \leq m)$, where β_i''s are formulas in the normal form, then there exists a formula β in the normal form such that $((\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)$ prj $\beta_0 \cong \beta$ $(1 \leq l \leq m)$. In the later proof, the axiom IUM will be used.

$$
\begin{aligned}
& (\beta_1, \ldots, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
\cong\ & (\beta_1, \ldots, \beta_i, \ldots, \beta_l, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee(\beta_1, \ldots, \beta_i \wedge \neg\varepsilon, \ldots, \beta_l, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee \bigvee_{t=i+1}^{l-1}(\beta_1, \ldots, \bigwedge_{h=i}^{t-1}\beta_h \wedge \varepsilon, \beta_t \wedge \neg\varepsilon, \ldots, \beta_l, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee(\beta_1, \ldots, \bigwedge_{h=i}^{l-1}\beta_h \wedge \varepsilon, \beta_l \wedge \neg\varepsilon, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0
\end{aligned}
$$

When $i = 1$ in IUM, that is, for any formula $((\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0$ $(1 \le l \le m)$, we have

$$
\begin{aligned}
& ((\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
\cong\ & (\beta_1, \ldots, \beta_l, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee(\beta_1 \wedge \neg\varepsilon, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee \bigvee_{t=2}^{l-1}(\bigwedge_{h=1}^{t-1}\beta_h \wedge \varepsilon, \beta_t \wedge \neg\varepsilon, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee(\bigwedge_{h=1}^{l-1}\beta_h \wedge \varepsilon, \beta_l \wedge \neg\varepsilon, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0
\end{aligned}
$$

According to the above deduction, $((\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0$ can be transformed into its normal form in the axiom system if and only if all the disjuncts on the right side can be transformed into its normal form by deduction. The first disjunct is a projection formula which is the case proved in case (5). The second disjunct can be further deduced:

$$
\begin{aligned}
& (\beta_1 \wedge \neg\varepsilon, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
\cong\ & (\bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \bigcirc\beta_{1i}, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ (\alpha_{0e} \wedge \varepsilon \vee \bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j}) && \text{TAU, Hypothesis} \\
\cong\ & \bigvee_{i=1}^{n_1}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ (\alpha_{0e} \wedge \varepsilon) \\
& \vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ (\alpha_{0j} \wedge \bigcirc\beta_{0j}) && \text{PDF, PDB} \\
\cong\ & \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \alpha_{0e} \wedge \bigcirc(\beta_{1i}; \ldots; \beta_l; (\beta_1; \ldots; \beta_l)^{+}; \ldots; \beta_m) \\
& \vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}\alpha_{1i} \wedge \alpha_{0j} \wedge \bigcirc(\beta_{1i}; (\beta_2, \ldots, \beta_l, (\beta_1, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_{0j}) && \text{PSB, PSF, IEC, PNX}
\end{aligned}
$$

Other disjuncts can also be further processed in a similar way, hence the origin formula can be transformed into its normal form in the proof system.

In step two, we need to prove that if $\beta_i \cong \beta_i'$ $(0 \le i \le m)$, where β_i''s are formulas in the normal form, then there exists a formula β in the normal form such that $(\beta_1, \ldots, (\beta_i, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \cong \beta$ $(1 < i \le l \le m)$. The proof proceeds by induction on i.

Base: $i = 2$, we have

$$
\begin{aligned}
& (\beta_1, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
\cong\ & (\alpha_{1e} \wedge \varepsilon \vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \bigcirc\beta_{1i}, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ (\alpha_{0e} \wedge \varepsilon \vee \bigvee_{j=1}^{n_0}\alpha_{0j} \wedge \bigcirc\beta_{0j}) && \text{Hypothesis} \\
\cong\ & (\alpha_{1e} \wedge \varepsilon, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \alpha_{0e} \wedge \varepsilon \\
& \vee \bigvee_{i=1}^{n_1}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \alpha_{0e} \wedge \varepsilon \\
& \vee \bigvee_{j=1}^{n_0}(\alpha_{1e} \wedge \varepsilon, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \alpha_{0j} \wedge \bigcirc\beta_{0j} \\
& \vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, (\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \alpha_{0j} \wedge \bigcirc\beta_{0j} && \text{PDF, PDB} \\
\cong\ & \alpha_{1e} \wedge \alpha_{0e} \wedge ((\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \varepsilon \\
& \vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \alpha_{0e} \wedge \bigcirc(\beta_{1i}; (\beta_2; \ldots; \beta_l)^{+}; \ldots; \beta_m) \\
& \vee \bigvee_{j=1}^{n_0}\alpha_{1e} \wedge \alpha_{0j} \wedge ((\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \bigcirc\beta_{0j} \\
& \vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}\alpha_{1i} \wedge \alpha_{0j} \wedge \bigcirc(\beta_{1i}; ((\beta_2, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_{0j}) && \text{PSM, PSF, PSB, IEC, PNX}
\end{aligned}
$$

According to the proof in step one, when $i = 2$, the above formula can be transformed into its normal form.

Induction: Suppose that when $i = k - 1$ the conclusion holds. When $i = k$ we have

$$
\begin{aligned}
& (\beta_1, \beta_2, \ldots, (\beta_k, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
\cong\ & (\alpha_{1e} \wedge \varepsilon, \beta_2, \ldots, (\beta_k, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee \bigvee_{i=1}^{n_1}(\alpha_{1i} \wedge \bigcirc\beta_{1i}, \beta_2, \ldots, (\beta_k, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 && \text{Hypothesis, PDF} \\
\cong\ & \alpha_{1e} \wedge (\beta_2, \ldots, (\beta_k, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_0 \\
& \vee \bigvee_{i=1}^{n_1}\alpha_{1i} \wedge \alpha_{0e} \wedge \bigcirc(\beta_{1i}; \beta_2; \ldots; (\beta_k; \ldots; \beta_l)^{+}; \ldots; \beta_m) \\
& \vee \bigvee_{i=1}^{n_1}\bigvee_{j=1}^{n_0}\alpha_{1i} \wedge \alpha_{0j} \wedge \bigcirc(\beta_{1i}; (\beta_2, \ldots, (\beta_k, \ldots, \beta_l)^{\oplus}, \ldots, \beta_m)\ prj\ \beta_{0j}) && \text{PSF, PSM, PSB, IEC, PNX, PDB}
\end{aligned}
$$

Hence, with the hypothesis and the proof given above, we prove that when $i = k$ the conclusion also holds.

(7) For *CCM*, refer to the proof of Lemma 4. □

Lemma 5. *For any terminable CCM–PPTL formula β, if $\nvdash \beta \rightarrow false$, then β is satisfiable.*

Proof. Since β is terminable, by Definition 6, $\beta \wedge \Diamond\varepsilon \not\equiv false$, which means that there exists a model σ that satisfies $\beta \wedge \Diamond\varepsilon$. Then σ is also a model of β and hence β is satisfiable. □

Lemma 6. *For any CCM–PPTL formula β and β', if $\beta \cong \beta'$ where β is non-terminable and β' in normal form, then β' must be of the form $\bigvee_{i=1}^{n} \beta_i \wedge \bigcirc\beta_i$ with each β_i being non-terminable.*

Proof. It is proved by contradiction.

(1) Suppose that $\beta \cong \alpha_e \wedge \varepsilon \vee \bigvee_{i=1}^{n} \alpha_i \wedge \bigcirc\beta_i$ and $\nvdash \alpha_e \rightarrow false$. For the completeness of the axiom system of the classical propositional logic, we have $\not\models \alpha_e \rightarrow false$, which means $\neg\alpha_e$ is not valid. That is, α_e is satisfiable. Therefore, $\alpha_e \wedge \varepsilon \wedge \Diamond\varepsilon \not\equiv false$ and moreover $\beta \wedge \Diamond\varepsilon \not\equiv false$, which is a contradiction to the premise that β is non-terminable. Hence, $\beta \cong \bigvee_{i=1}^{n} \alpha_i \wedge \bigcirc\beta_i$.

(2) Suppose that there exists a terminable formula β_i. We can easily derive that β is also terminable, which is a contradiction to the premise. Hence each β_i is non-terminable. □

Lemma 6 tells us that if a non-terminable formula β has been deduced into its normal form β' using the axiom system, then we can infer that there is no terminal product $\alpha_e \wedge \varepsilon$ in β', and each future product in its normal form is also non-terminable. Otherwise, there is a contradiction to the premise that β is non-terminable.

To prove Lemma 7, we need Tarski's fix-point theorem, Scott's fix-point induction and the definition of inclusion subset which are presented in Theorem 3, Theorem 4 and Definition 7.

Theorem 3 *(Tarski's fix-point theorem). Every monotonic function F over a complete lattice $\langle B, \sqsubseteq\rangle$ has a unique least fix point $\sqcup_{n\in N_0} F^n(\bot)$ and a unique greatest fix point $\sqcap_{n\in N_0} F^n(\top)$.*

Definition 7 *(Inclusion subset).* Let P be a complete partial order. A subset D of P is inclusive iff for all ω-chains $d_0 \sqsubseteq d_1 \ldots \sqsubseteq d_n \sqsubseteq \ldots$ in P if $d_n \in D$ for all $n \in N_0$ then $\bigsqcup_{n\in N_0} d_n \in D$.

Theorem 4 *(Scott's fix-point induction). Let B be a complete partial order with a bottom ($\bot$), $F : B \rightarrow B$ a continuous function, and D an inclusion subset of B. If $\bot \in D$ and $\forall x \in B.\ x \in D \rightarrow F(x) \in D$, then $fix(F) \in D$.*

Lemma 7. *For any non-terminable CCM–PPTL formula β satisfying $\nvdash \beta \rightarrow false$, β is satisfiable.*

Proof. To prove the non-terminable formula β is satisfiable, we need to construct an infinite interval σ for β and then to prove the interval σ is just a model of β.

By Lemma 6 and β being non-terminable, we have $\beta \cong \bigvee_{i=1}^{n} \alpha_i \wedge \bigcirc\beta_i$ and each β_i being non-terminable. Since $\nvdash \beta \rightarrow false$, we can deduce that $\nvdash \beta_i \rightarrow false$ for some β_i. This deduction is formalized as follows:

	$\nvdash \beta \rightarrow false$ and β is non-terminable	Premise
$\Longrightarrow$	$\nvdash \bigvee_{i=1}^{n} \alpha_i \wedge \bigcirc\beta_i \rightarrow false$	Lemma 6
$\Longrightarrow$	$\nvdash \alpha_i \wedge \bigcirc\beta_i \rightarrow false$ for some i	TAU
$\Longrightarrow$	$\nvdash \alpha_i \rightarrow false$ and $\nvdash \bigcirc\beta_i \rightarrow false$	TAU
$\Longrightarrow$	there is a model $\langle s_0\rangle$ satisfies α_i and $\nvdash \beta_i \rightarrow false$	Completeness of PL

We use s_0 as the first state of the interval σ being constructed. Through applying a similar deduction to β_i, we can generate another state s_1 which is treated as the second state of σ. Then repeat this procedure infinite times since β is non-terminable, we get an infinite interval $\langle s_0, s_1, s_2, \ldots\rangle$ which is denoted by σ. Now we need to prove σ is a model of β.

First, some notations are introduced. Let $\sigma^{-1} = \emptyset$, $\sigma^i = \langle s_0 \ldots, s_i\rangle$ $(i \in N_0)$, and $\sigma^\omega = \lim_{i\rightarrow\infty} \sigma^i = \sigma = \langle s_0, s_1, \ldots\rangle$. Define $B = \{\sigma^i | i \in \{-1\} \cup N_\omega\}$ with the *Prefix* relation which is defined as follows: $Prefix(\sigma^i, \sigma^j)$ is true iff $i \leq j$. Let $F : B \rightarrow B$ be a function given by $F(\sigma^i) = \sigma^{i+1}$ and $F(\sigma^\omega) = \sigma^\omega$.

First, we need to prove that $(B, Prefix)$ is a complete partial order.

(1) Reflexivity: for all $\sigma^i \in B$, we have $Prefix(\sigma^i, \sigma^i)$ is true.
(2) Anti-symmetry: if $Prefix(\sigma^i, \sigma^j)$ and $Prefix(\sigma^j, \sigma^i)$ hold, then we have $i \leq j$ and $j \leq i$, leading to $i = j$. Hence, $\sigma^i = \sigma^j$.
(3) Transitivity: if $Prefix(\sigma^i, \sigma^j)$ and $Prefix(\sigma^j, \sigma^k)$ hold, then we have $i \leq j \leq k$, so $Prefix(\sigma^i, \sigma^k)$ is true.

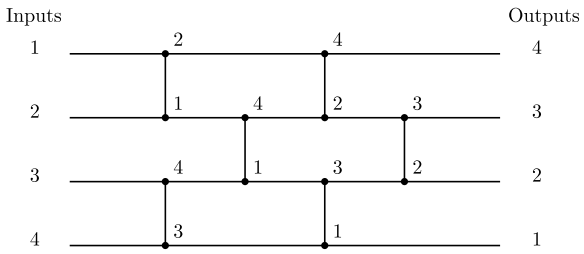

Fig. 2. The odd–even sorting network(1).

Hence, $(B, Prefix)$ is a partial order. Further, there exist a least upper bound (lub) and a greatest lower bound (glb) in every non-empty subset $A = \{\sigma^{i_1}, \ldots, \sigma^{i_n}\}$ of B, where $-1 \leq i_1 \leq \ldots \leq i_{n-1} \preceq i_n \leq \omega$. The greatest lower bound is $glb(A) = \sigma^{i_1}$. If A is a finite set, the least upper bound is $lub(A) = \sigma^{i_n}$; otherwise, the least upper bound is $lub(A) = \sigma^{\omega}$. Hence, $(B, Prefix)$ is a complete partial order and also a complete lattice.

Then, we need to prove that the function F is continuous.

Suppose that $Prefix(\sigma^i, \sigma^j)$ is true, then $i \leq j$, and so $i+1 \leq j+1$. As a result, $Prefix(F(\sigma^i), F(\sigma^j))$. Hence F is monotonic. Furthermore, let σ^{i_0} *Prefix* σ^{i_1} be the infix form of $Prefix(\sigma^{i_0}, \sigma^{i_1})$. Suppose that σ^{i_0} *Prefix* σ^{i_1} *Prefix* $\sigma^{i_2} \ldots$ is an arbitrary ω-chain in B. Then, we have

$$\sqcup_{n\in N_0} F(\sigma^{i_n}) = \sqcup_{n\in N_0} \sigma^{i_{n+1}} = \sigma^{\omega} = F(\sigma^{\omega}) = F(\sqcup_{n\in N_0} \sigma^{i_n})$$

Hence, F is continuous. Since $(B, Prefix)$ is a complete lattice and the function F is continuous, we can obtain the least fix-point of F by Tarski's fix-point theorem:

$$fix(F) = \sqcup_{n\in N_0} F^n(\sigma^{-1}) = \sigma^{\omega}$$

where σ^{ω} denotes the whole state sequence σ. Now we construct a subset D of B, as follows:

$$D = \{\sigma^i | \sigma^i \in B \text{ and } \sigma^i \text{ is a prefix of a model of } \beta\}$$

By induction on the length of the prefix of the interval σ, each finite prefix of σ is proved to be a prefix of the final model. Hence, for any ω-chain in B

$$\sigma^{i_0} \; Prefix \; \sigma^{i_1} \; Prefix \; \cdots$$

it is in fact in D since $\sigma^i \in D$ for any $i \in N_0$. Hence, $\sigma^{\omega} \in D$, namely, $\sqcup_{n\in N_0} \in D$. Therefore, D is an inclusion subset of B. In addition, $\emptyset \in D$. Then by Scott's fix-point induction, it's readily to prove that the infinite prefix of σ is also a prefix of the final model, which means that σ is just a model of β. Therefore, β is satisfiable. □

The proof of Lemma 7 is a little complicated. It involves constructing an interval for β and then prove the interval is indeed a model of β. Two famous theorems on fix-point are used in the proof, one is Tarski's fix-point theorem and the other is Scott's fix-point induction. From Lemma 5 and 7, we can derive that any CCM–PPTL formula β, no matter it is terminable or non-terminable, if $\nvdash \beta \rightarrow false$, then β is satisfiable. Thus, we have the following completeness theorem.

Theorem 5 *(Completeness). For any CCM–PPTL formula β, if $\models \beta$, then $\vdash \beta$.*

Proof. From the premise of β is valid, we can derive the duality that $\neg\beta$ is unsatisfiable. By Lemma 5 and 7, we have $\vdash \neg\beta \rightarrow false$ is a theorem in the proof system of CCM–PPTL, which means that $\vdash \beta$ is a theorem in the proof system. □

Now, we have proved that the extended axiom system including CCM constructs preserve the soundness and completeness of the original one. Then, an example is given to illustrate how to use the proof system to reason out a property of an odd–even sorting algorithm.

5. Example

In this section, an odd–even sorting network [12], shown in Fig. 2, is modeled and verified using Cylinder Computation Model. An odd–even sorting network is a relatively simple sorting algorithm, developed originally for use on parallel processors with local interconnections. It can also be used as a parallel sorting algorithm on multi-core processors. It functions by comparing all (even, odd)-indexed pairs of adjacent elements in the list (or array). If a pair is in the wrong order (the first is larger than the second) the elements are swapped. The next step repeats this process for (odd, even)-indexed pairs (of adjacent elements). Thus, it alternates between (even, odd) and (odd, even) steps until the list is sorted.

In Fig. 2, each vertical line means a comparison between its two inputs. This operation is performed by a kind of computing element called comparator with two input wires and two output wires. The comparator forwards the larger to its top

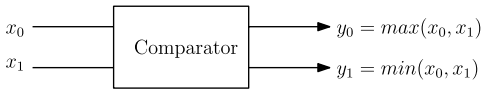

Fig. 3. A comparator.

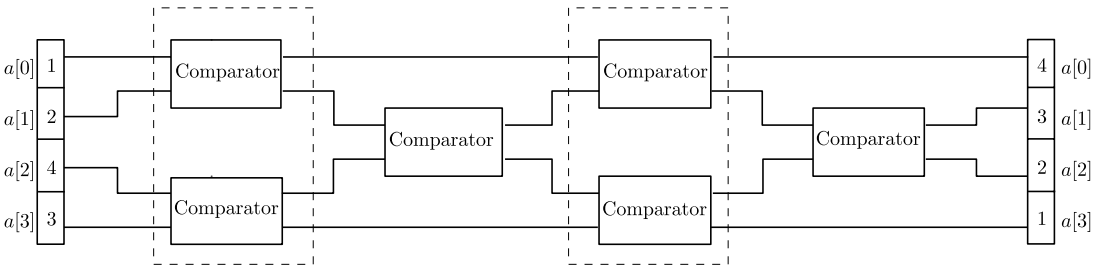

Fig. 4. The odd–even sorting network(2).

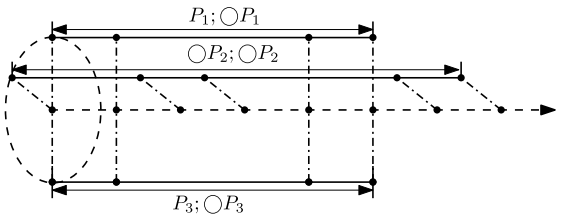

Fig. 5. The CCM of the odd–even sorting network.

wire and the smaller to its bottom wire (see Fig. 3). Actually, the odd–even sorting network can be implemented in Fig. 4, which indicates that the comparison operations in one dashed frame can be performed in parallel, and the comparisons in the horizontal direction are performed sequentially. If an array $a[\,] = \{1, 2, 4, 3\}$ is transmitted as an input for the network, after computing, the elements in the array will be reassigned in a descending order.

We can model this algorithm using CCM as follows, where P_1 is delegated to compare elements $a[0]$ and $a[1]$; P_2 to compare elements $a[1]$ and $a[2]$; P_3 to compare elements $a[2]$ and $a[3]$. The cylinder computation model of the algorithm is formalized as *OddEven*. The projected intervals over which P_1, P_2 and P_3 are interpreted are depicted in Fig. 5.

$$
\begin{aligned}
P_1 &\stackrel{\text{def}}{=} \texttt{if } (a[0] < a[1]) \texttt{ then } (a[0] := a[1] \texttt{ and } a[1] := a[0]) \texttt{ else } \mathsf{skip} \\
P_2 &\stackrel{\text{def}}{=} \texttt{if } (a[1] < a[2]) \texttt{ then } (a[1] := a[2] \texttt{ and } a[2] := a[1]) \texttt{ else } \mathsf{skip} \\
P_3 &\stackrel{\text{def}}{=} \texttt{if } (a[2] < a[3]) \texttt{ then } (a[2] := a[3] \texttt{ and } a[3] := a[2]) \texttt{ else } \mathsf{skip} \\
\mathit{OddEven} &\stackrel{\text{def}}{=} \mathsf{frame}(a) \texttt{ and } a[\,] = \{1, 2, 4, 3\} \texttt{ and} \\
&\quad (P_1; \bigcirc P_1 \texttt{ ov } (1 \cdot 3 \cdot 1) \| \bigcirc P_2; \bigcirc P_2 \texttt{ ov } (2 \cdot 1 \cdot 3 \cdot 1) \| P_3; \bigcirc P_3 \texttt{ ov } (1 \cdot 3 \cdot 1))
\end{aligned}
$$

Further, we can verify a property of the algorithm using the proof system of CCM–PPTL. The property is formalized as follows:

$$\mathsf{fin}(0 \le i \le j \le 3 \rightarrow a[i] \ge a[j])$$

The proof of the property is given as follows:

(1)	$\mathit{OddEven} \supset ((a[0] := a[1] \texttt{ and } a[1] := a[0]); \bigcirc P_1) \texttt{ ov } (1 \cdot 3 \cdot 1)$	TAU
	$\| \bigcirc (P_2; \bigcirc P_2) \texttt{ ov } (2 \cdot 1 \cdot 3 \cdot 1)$	PNX
	$\|(\mathsf{skip}; \bigcirc P_3) \texttt{ ov } (1 \cdot 3 \cdot 1)$	TAU
(2)	$\mathit{OddEven} \supset ((a[0] := 2 \texttt{ and } a[1] := 1); \bigcirc P_1) \texttt{ ov } (1 \cdot 3 \cdot 1)$	TAU
	$\| \bigcirc (P_2; \bigcirc P_2) \texttt{ ov } (2 \cdot 1 \cdot 3 \cdot 1)$	
	$\|(\bigcirc \varepsilon; \bigcirc P_3) \texttt{ ov } (1 \cdot 3 \cdot 1)$	Def of skip
(3)	$\mathit{OddEven} \supset (\bigcirc (a[0] = 2 \texttt{ and } a[1] = 1); \bigcirc P_1) \texttt{ ov } (1 \cdot 3 \cdot 1)$	Def of :=
	$\| \bigcirc (P_2; \bigcirc P_2) \texttt{ ov } (2 \cdot 1 \cdot 3 \cdot 1)$	
	$\| \bigcirc \bigcirc P_3 \texttt{ ov } (1 \cdot 3 \cdot 1)$	PNX, PEE
(4)	$\mathit{OddEven} \supset \bigcirc ((a[0] = 2 \texttt{ and } a[1] = 1; \bigcirc P_1) \texttt{ ov } (3 \cdot 1))$	A4
	$\| \bigcirc^2 ((P_2; \bigcirc P_2) \texttt{ ov } (1 \cdot 3 \cdot 1))$	A4
	$\| \bigcirc (\bigcirc P_3 \texttt{ ov } (3 \cdot 1))$	A4

$$
\begin{array}{lll}
(5) & \mathit{OddEven} \supset \bigcirc(a[0]=2 \wedge a[1]=1 \wedge a[2]=4 \wedge a[3]=3 & \text{A5}\\
& \quad \wedge (\bigcirc P_1 \ \mathtt{ov}\ (3\cdot 1) & \\
& \quad\quad \| \bigcirc (P_2; \bigcirc P_2) \ \mathtt{ov}\ (1\cdot 3\cdot 1) & \\
& \quad\quad \| \bigcirc P_3 \ \mathtt{ov}\ (3\cdot 1))) & \\
(6) & \mathit{OddEven} \supset \bigcirc^2(a[0]=2 \wedge a[1]=1 \wedge a[2]=4 \wedge a[3]=3 & \text{A5}\\
& \quad \wedge (\bigcirc^2(P_1 \ \mathtt{ov}\ (1)) & \text{A4}\\
& \quad\quad \|(P_2; \bigcirc P_2) \ \mathtt{ov}\ (1\cdot 3\cdot 1) & \text{A4}\\
& \quad\quad \| \bigcirc^2 (P_3 \ \mathtt{ov}\ (1)))) & \text{A4}\\
(7) & \mathit{OddEven} \supset \bigcirc^2(a[0]=2 \wedge a[1]=1 \wedge a[2]=4 \wedge a[3]=3 & \\
& \quad \wedge (\bigcirc^2(P_1 \ \mathtt{ov}\ (1)) & \\
& \quad\quad \|(a[1]:=a[2] \ \mathtt{and}\ a[2]:=a[1]; \bigcirc P_2) \ \mathtt{ov}\ (1\cdot 3\cdot 1) & \text{TAU}\\
& \quad\quad \| \bigcirc^2 (P_3 \ \mathtt{ov}\ (1)))) & \\
(8) & \mathit{OddEven} \supset \bigcirc^2(a[0]=2 \wedge a[1]=1 \wedge a[2]=4 \wedge a[3]=3 & \\
& \quad \wedge (\bigcirc^2(P_1 \ \mathtt{ov}\ (1)) & \\
& \quad\quad \|(a[1]:=4 \ \mathtt{and}\ a[2]:=1; \bigcirc P_2) \ \mathtt{ov}\ (1\cdot 3\cdot 1) & \text{TAU}\\
& \quad\quad \| \bigcirc^2 (P_3 \ \mathtt{ov}\ (1)))) & \\
(9) & \mathit{OddEven} \supset \bigcirc^4(a[0]=2 \wedge a[1]=4 \wedge a[2]=1 \wedge a[3]=3 & \\
& \quad \wedge ((a[1]:=a[0] \ \mathtt{and}\ a[0]:=a[1]) \ \mathtt{ov}\ (1) & \text{TAU}\\
& \quad\quad \| \bigcirc^2 (P_2 \ \mathtt{ov}\ (1)) & \\
& \quad\quad \|(a[2]:=a[3] \ \mathtt{and}\ a[3]:=a[2]) \ \mathtt{ov}\ (1))) & \text{TAU}\\
(10) & \mathit{OddEven} \supset \bigcirc^4(a[0]=2 \wedge a[1]=4 \wedge a[2]=1 \wedge a[3]=3 & \\
& \quad \wedge ((a[0]:=4 \ \mathtt{and}\ a[1]:=2) \ \mathtt{ov}\ (1) & \text{TAU}\\
& \quad\quad \| \bigcirc^2 (P_2 \ \mathtt{ov}\ (1)) & \\
& \quad\quad \|(a[2]:=3 \ \mathtt{and}\ a[3]:=1) \ \mathtt{ov}\ (1))) & \text{TAU}\\
(11) & \mathit{OddEven} \supset \bigcirc^4(a[0]=2 \wedge a[1]=4 \wedge a[2]=1 \wedge a[3]=3 & \\
& \quad \wedge (\bigcirc(a[0]=4 \wedge a[1]=2) & \text{A4}\\
& \quad\quad \| \bigcirc^2 (P_2 \ \mathtt{ov}\ (1)) & \\
& \quad\quad \| \bigcirc (a[2]=3 \wedge a[3]=1))) & \text{TAU}\\
(12) & \mathit{OddEven} \supset \bigcirc^6(a[0]=4 \wedge a[1]=2 \wedge a[2]=3 \wedge a[3]=1 & \\
& \quad \wedge (P_2 \ \mathtt{ov}\ (1))) & \text{TAU, PNX}\\
(13) & \mathit{OddEven} \supset \bigcirc^6(a[0]=4 \wedge a[1]=2 \wedge a[2]=3 \wedge a[3]=1 & \\
& \quad \wedge ((a[1]:=a[2] \ \mathtt{and}\ a[2]:=a[1]) \ \mathtt{ov}\ (1))) & \text{TAU}\\
(14) & \mathit{OddEven} \supset \bigcirc^6(a[0]=4 \wedge a[1]=2 \wedge a[2]=3 \wedge a[3]=1 & \\
& \quad \wedge ((a[1]:=3 \ \mathtt{and}\ a[2]:=2) \ \mathtt{ov}\ (1))) & \text{TAU}\\
(15) & \mathit{OddEven} \supset \bigcirc^6(a[0]=4 \wedge a[1]=2 \wedge a[2]=3 \wedge a[3]=1 & \\
& \quad \wedge (\bigcirc(a[1]=3 \wedge a[2]=2))) & \text{A4}\\
(16) & \mathit{OddEven} \supset \bigcirc^7(a[0]=4 \wedge a[1]=3 \wedge a[2]=2 \wedge a[3]=1 \wedge \varepsilon) & (15)\\
(17) & \mathit{OddEven} \supset \mathsf{fin}(0 \le i \le j \le 3 \rightarrow a[i] \ge a[j]) & (16)
\end{array}
$$

Note that only a small array is used here to illustrate the principle of proofs for using the CCM–PPTL axiom system. As for a large array with w elements, a Bitonic sorting network [12] can be employed. The modeling and proving processes are similar.

6. Conclusion

We introduce a Cylinder Computation Model into Propositional Projection Temporal Logic and present an axiom system for CCM–PPTL, which can be employed to model, specify and verify multi-core computation systems. In the future, we need to do some further case studies for complex multi-core computations. Further, to provide a highly automatic verification approach, the existing tool for PPTL theorem prover will be extended to support CCM construct. Moreover, we will explore the verification methodology which combines model checking and theorem proving.

References

[1] D. Arden, Delayed-logic and finite-state machines, in: Theory of Computing Machine Design, Univ. of Michigan Press, 1960, pp. 1–35.
[2] Y. Bertot, P. Castéran, Interactive Theorem Proving and Program Development, Heidelberg, 2004.
[3] W. Bledsoe, D. Loveland, Automating Theorem Proving: After 25 Years, Providence, 1984.
[4] B. Brock, M. Kaufmann, J. Moore, ACL2 theorems about commercial microprocessors, in: M. Srivas, A. Camilleri (Eds.), Proceedings of the First International Conference on Formal Methods in Computer-Aided Design, in: LNCS, vol. 1166, Springer, London, 1996, pp. 275–293.
[5] E. Clarke, E. Emerson, Design and synthesis of synchronisation skeletons using branching time temporal logic, in: D. Kozen (Ed.), Proceedings of the Workshop on Logic of Programs, in: LNCS, vol. 131, Springer, Heidelberg, 1981, pp. 52–71.
[6] Z. Duan, Temporal Logic and Temporal Logic Programming, Science Press, Beijing, 2005.

[7] Z. Duan, C. Tian, A unified model checking approach with projection temporal logic, in: Proceedings of ICFEM 2008, 2008, pp. 167–186.
[8] Z. Duan, C. Tian, A practical decision procedure for propositional projection temporal logic with infinite models, Theoret. Comput. Sci. 554 (2014) 169–190, http://dx.doi.org/10.1016/j.tcs.2014.02.011.
[9] Z. Duan, C. Tian, L. Zhang, A decision procedure for propositional projection temporal logic with infinite models, Acta Inform. 45 (1) (2008) 43–78.
[10] Z. Duan, N. Zhang, M. Koutny, A complete proof system for propositional projection temporal logic, Theoret. Comput. Sci. 497 (2013) 84–107.
[11] M. Gordon, T. Melham, Introduction to HOL: A Theorem Proving Environment for Higher Order Logic, Cambridge University Press, 1993.
[12] M. Herlihy, N. Shavit, The Art of Multiprocessor Programming, Morgan Kaufmann Publishers, 2008.
[13] G. Holzmann, The model checker SPIN, IEEE Trans. Softw. Eng. 23 (5) (1997) 279–295.
[14] K. McMillan, Symbolic Model Checking: An Approach to the State Explosion Problem, Dordrecht, 1993.
[15] R. Milner, Communicating and Mobile System: The Π-Calculus, Cambridge University Press, Cambridge, 1999.
[16] S. Owre, J. Rushby, N. Shankar, A prototype verification system, in: PVS: Proceedings of the 11th International Conference on Automated Deduction, in: LNAI, vol. 607, Springer, Heidelberg, 1992, pp. 748–752.
[17] L. Paulson, Isabelle – a generic theorem prover, LNCS 828 (1994).
[18] J. Queille, J. Sifakis, Specification and verification of concurrent systems in CESAR, in: M. Dezani-Ciancaglini, U. Montanari (Eds.), Proceedings of the 5th Colloquium on International Symposium in Programming, in: LNCS, vol. 137, Springer, London, 1982, pp. 337–351.
[19] A. Sistla, Theoretical issues in the design and verification of distributed systems, Ph.D. Thesis, Harvard University, 1983.
[20] C. Tian, Z. Duan, Expressiveness of propositional projection temporal logic with star, Theoret. Comput. Sci. 412 (18) (2011) 1729–1744.
[21] M. Vardi, A temporal fixpoint calculus, in: POPL'88, 1988, pp. 250–259.
[22] P. Wolper, Temporal logic can be more expressive, Inf. Control 56 (1983) 72–99.
[23] N. Zhang, Z. Duan, C. Tian, A cylinder computation model for many-core parallel computing, Theoret. Comput. Sci. 497 (2013) 68–83.