TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 13/18 pp372–385
DOI: 10.26599/TST.2020.9010054
Volume 27, Number 2, April 2022

An Axiom System of Probabilistic Mu-Calculus

Wanwei Liu*, Junnan Xu, David N. Jansen, Andrea Turrini, and Lijun Zhang

Abstract: Mu-calculus (a.k.a. μTL) is built up from modal/dynamic logic via adding the least fixpoint operator μ. This type of logic has attracted increasing attention since Kozen's seminal work. PμTL is a succinct probabilistic extension of the standard μTL obtained by making the modal operators probabilistic. Properties of this logic, such as expressiveness and satisfiability decision, have been studied elsewhere. We consider another important problem: the axiomatization of that logic. By extending the approaches of Kozen and Walukiewicz, we present an axiom system for PμTL. In addition, we show that the axiom system is complete for aconjunctive formulas.

Key words: PμTL; axiom system; aconjunctive formula; tableau approach

1 Introduction

In Ref. [1], Kozen presented propositional μ-calculus, (a.k.a. μTL), which is an extension of modal logic with the least fixed point operator. μTL has been proved to be such an important mathematical tool that it is widely used in reasoning and verification. Kozen also investigated an axiom system for μTL, and showed its completeness for *aconjunctive formulas*. That is, the negations of all unsatisfiable aconjunctive formulas are provable. However, completeness for the full logic is considerably intricate. After a decade, such property was finally shown in Ref. [2] by Walukiewicz based on the basis of the deep investigations given in Refs. [3–5]. Later, Tamura[6] reported an easy proof by introducing the notion of wide tableaux and by providing an alternative definition of a tableau consequence.

Probabilistic extensions of modal/temporal logics are highly important in both theory and practice, particularly in verification and game theory[7]. For example, the semantics of probabilistic programs were studied by Kozen in Ref. [8]. PCTL, which is the probabilistic version of CTL, was introduced in Ref. [9], and it has become the most important specification language in probabilistic model checking. In addition, the logic PCTL* subsumes PCTL and PLTL, whose model checking and satisfiability problem have been intensively studied[10–14]. In Ref. [15], a deductive approach for PCTL* was presented, but it was incomplete.

In Ref. [16], it is shown that some crucial properties used in probabilistic systems (e.g., Ref. [17]), such as the "safety property", cannot be captured by PCTL. Actually, a desirable system is one that can guarantee with a large probability that error does not occur. However, from a perspective described by PCTL, such system leads to error with probability 1 in the long run, and is thus counterintuitive; hence, no quantitative description of safety can be yielded by such logic.

Various probabilistic extensions of μTL have also been investigated[4, 18–21]. In Ref. [22], Mio presented an extension of μ-calculus by interpreting a formula into a function from states to [0, 1]. The extension could encode full PCTL, but its model checking and satisfiability

• Wanwei Liu is with College of Computer Science, National University of Defense Technology, Changsha 410073, China. E-mail: wwliu@nudt.edu.cn.
• Junnan Xu, David N. Jansen, and Andrea Turrini are with State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100039, China. E-mail: {xujn, dajansen, turrini}@ios.ac.cn.
• Lijun Zhang is with State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China; University of Chinese Academy of Sciences, Beijing 100039, China; and also with Institute of Intelligent Software, Guangzhou 511458, China. E-mail: zhanglj@ios.ac.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2020-09-24; revised: 2020-10-15; accepted: 2020-10-19

algorithms are open. In Ref. [23], another probabilistic μ-calculus was introduced. It also subsumes PCTL formulas, but it only involves alternation-free formulas.

Direct extensions of μTL were introduced independently in Refs. [16, 24]. The logic PμTL in Ref. [16] uses the probabilistic "next" operator $\mathsf{X}^{\gtrsim p}$ to replace the modal operators $\Diamond$ and operator $\Box$. Indeed, the modalities $\Diamond$ and $\Box$ can be represented by $\mathsf{X}^{>0}$ and $\mathsf{X}^{\geqslant 1}$, respectively. The logic suggested in Ref. [24] subsumes that in Ref. [16], but its satisfiability problem remains open. By contrast, the satisfiability problem of PμTL was proved to be decidable in **2EXPTIME** in Ref. [16]. In Ref. [25], the problem was found to be **EXPTIME**-complete. In Ref. [26], PμTL was further extended into the alternating-time case. In Ref. [27], another probabilistic μ-calculus is presented, and that logic allows the encoding of inequational conditions. The authors also presented an axiom system, which is sound and complete for alternation-free formulas[27]. Such axiom system includes 9 axioms and 2 inference rules for basic formulas, together with 3 inference rules for maximal equation blocks and 3 inference rules for minimum blocks.

In the current work, we intend to establish a Kozen-style axiom system for PμTL. Our approach is based on Refs. [1, 2]. The axiom schemes such as $\langle a\rangle f \vee \langle a\rangle g \leftrightarrow \langle a\rangle(f \vee g)$ and $\langle a\rangle f \wedge [a]g \to \langle a\rangle(f \wedge g)$ suffice for standard modal logic, but the same is not true in the probabilistic setting. For example, these schemes cannot prove probability deductions such as

$$\mathsf{X}^{>0.7} f \wedge \mathsf{X}^{>0.7} g \to \mathsf{X}^{>0.4}(f \wedge g).$$

To show the completeness of Kozen's axiom system for μTL, Walukiewicz[2] first established a tableau/refutation-based approach (which was first suggested by Streett and Emerson[28]), and reduced the satisfiability problem of μTL to a special game upon the tableau. By observing the fact that if a formula has a thin refutation then its negation is provable, he showed the completeness for aconjunctive formulas[2]. In addition, he showed the completeness for the whole logic via defining the so-called tableau-consequence[2].

We add an axiom (PROB) and an inference rule (COV) to Kozen's system. Then, we adopt Kozen's approach to our axiom system and show the completeness of PμTL on aconjunctive formulas. Tableaux/refutations are also adapted to match the probabilistic settings. We show that the existence of a consistent tableau implies the satisfiability of the corresponding formula. We also reveal that an unsatisfiable (i.e., inconsistent) formula must have a feasible refutation. Then we prove that if a PμTL formula has a thin refutation then its negation must be provable. Extending the completeness proof to all formulas (i.e., non-acoujunctive formulas) is quite involved, and the detailed reasons are analyzed in Section 6. We leave this part as our future work.

This paper is an extension and deep revision of the conference version[29]. In comparison to the conference version, we do not use the game-based definitions on tableaux and/or refutations to make these notions more intuitive and succinct. Meanwhile, we have added as new material illustrations how to use the presented axiom system, but more importantly, we also provide a new proof of the main completeness theorem (Theorem 10 in this paper), and we fixed some imperfections of the original proof.

The remainder of this paper is organized as follows. In Section 2, we revisit some related notions, including PμTL, automata, and games. In Section 3, we introduce tableaux for PμTL. And in Section 4, we present an axiom system of PμTL. In Section 5, we prove the soundness and completeness of the presented axiom system for aconjunctive formulas. In Section 6, we discuss the technical difficulties when we try to adapt Walukiewicz's proof to the full logic PμTL.

2 Preliminary

2.1 Logic PμTL

We first recall the logic PμTL defined in Ref. [16]. To define the syntax of this logic, we need to fix a countable set $\mathcal{A}$ of *atomic propositions* (the elements of which are typically a, a_1, a_2, etc.) and a countable set $\mathcal{Z}$ of *variables* (such as Z, Z_1, Z_2, etc.). In addition, we use p, p_1, p_2, etc. to denote probabilities.

PμTL formulas, ranging over f, g, etc., can be described by BNF as follows:

$$f ::= a \in \mathcal{A} \mid Z \in \mathcal{Z} \mid \neg f \mid f \vee g \mid \mathsf{X}^{>p} f \mid \mu Z.f.$$

For the formulas with the form $\mu Z.f$ we require all free occurrences of Z in f to be *positive*, i.e., within the scope of an even number of negations.

For a variable $Z \in \mathcal{Z}$, we regard an occurrence of it in f as *bound* if it is in some subformula $\mu Z.g$; otherwise, this occurrence is *free*.

We call a formula involving no free variables a *closed formula*. Given two formulas f and g, a variable Z, we denote by $f[g/Z]$ the formula obtained from f via replacing each free occurrence of Z in f by g.

We also define the derived operators $\top$, $\bot$, $\wedge$, $\mathsf{X}^{\geqslant p}$, and ν as follows:

$$\top \overset{\text{def}}{=} a \vee \neg a,$$
$$\bot \overset{\text{def}}{=} \neg\top,$$
$$f \wedge g \overset{\text{def}}{=} \neg(\neg f \vee \neg g),$$
$$\mathsf{X}^{\geqslant p} f \overset{\text{def}}{=} \neg\mathsf{X}^{>1-p}\neg f,$$
$$\nu Z.f \overset{\text{def}}{=} \neg\mu Z.\neg f[\neg Z/Z].$$

For convenience, we use the symbol $\gtrsim$ to range over $\{>, \geqslant\}$, and we use $\not\gtrsim$ to represent the other inequality sign, i.e., $\not\gtrsim \in \{>, \geqslant\} \setminus \{\gtrsim\}$. In addition, we use σ to range over $\{\mu, \nu\}$ and let $\overline{\sigma}$ be its dual, that is, $\overline{\sigma} = \mu$ if $\sigma = \nu$ and $\overline{\sigma} = \nu$ if $\sigma = \mu$. We also use the usual definition for $\bot$ and $\top$.

A *Markov chain* is a tuple $M = (S, P, L)$, where S is a set of *states* that may be infinite; $P\colon S \times S \to [0, 1]$ is a *distribution function*, satisfying $\sum_{s' \in S} P(s, s') = 1$ for every $s \in S$; and $L\colon S \to 2^{\mathcal{A}}$ is the *labeling function*. For each $s \in S$, we call the tuple (M, s) a *pointed Markov chain*.

The *Semantics* of a PμTL formula f can be given w.r.t. a Markov chain $M = (S, P, L)$ and an *evaluation* $\mathcal{E}\colon \mathcal{Z} \to 2^S$. A formula can capture a subset of S satisfying f, and we denoted it by $[\![f]\!]_M^{\mathcal{E}}$, which can be inductively defined as follows:

- $[\![a]\!]_M^{\mathcal{E}} = \{ s \in S \mid a \in L(s) \}$;
- $[\![Z]\!]_M^{\mathcal{E}} = \mathcal{E}(Z)$;
- $[\![\neg f]\!]_M^{\mathcal{E}} = S \setminus [\![f]\!]_M^{\mathcal{E}}$;
- $[\![f \vee g]\!]_M^{\mathcal{E}} = [\![f]\!]_M^{\mathcal{E}} \cup [\![g]\!]_M^{\mathcal{E}}$;
- $[\![\mathsf{X}^{>p} f]\!]_M^{\mathcal{E}} = \{ s \in S \mid \sum_{s' \in [\![f]\!]_M^{\mathcal{E}}} P(s, s') > p \}$;
- $[\![\mu Z.f]\!]_M^{\mathcal{E}} = \bigcap \{ S' \subseteq S \mid [\![f]\!]_M^{\mathcal{E}[Z \mapsto S']} \subseteq S' \}$,

where $\mathcal{E}[Z \mapsto S']$ is an evaluation that agrees with $\mathcal{E}$ except that it assigns S' to Z.

For convenience, we sometimes directly write $s \in [\![f]\!]_M^{\mathcal{E}}$ as $M, s \models_{\mathcal{E}} f$. When f is closed, we may abbreviate it to $M, s \models f$. We sometimes drop off the subscripts/superscripts from the semantics notation (i.e., $[\![\cdot]\!]$), when we are not concerned about the model or evaluation.

We say that f is *satisfiable* (resp. *valid*) if $M, s \models_{\mathcal{E}} f$ holds for some (resp. every) Markov chain M, some (resp. each) state s of M, and some (resp. every) evaluation $\mathcal{E}$. A formula f is *unsatisfiable* (or, inconsistent) if $\neg f$ is valid. We write $\models f$ if f is a valid formula.

As shown in Ref. [16], for PμTL, we have the following results on the expressiveness and satisfiability decision:

- PμTL and PCTL are incompatible in expressiveness, that is, there exist some properties that can only be expressed by one logic, and not by another.
- Once we are concerned about the qualitative fragment, that is, formulas use only the extreme probabilities 0 and 1, then qualitative PCTL is a proper fragment of qualitative PμTL if we further confine the models to finite ones.
- The satisfiability problem of PμTL is decidable. In Ref. [25], it is proved that this problem is **EXPTIME**-complete. Furthermore, PμTL formulas enjoy the *finite-model* property, that is, a formula is satisfiable if and only if it is satisfied by some finite Markov chain.

2.2 Binders and expansions

In this section, we revisit some notions relative to tableaux, that were originally presented for μTL. All these notions are mainly syntactic, and can thus be extended to PμTL.

By using the following rewriting rules,

$$\neg\neg f \Rightarrow f,$$
$$\neg(f \wedge g) \Rightarrow \neg f \vee \neg g,$$
$$\neg(f \vee g) \Rightarrow \neg f \wedge \neg g,$$
$$\neg\mathsf{X}^{\gtrsim p} f \Rightarrow \mathsf{X}^{\not\gtrsim 1-p}\neg f,$$
$$\neg\sigma Z.f \Rightarrow \overline{\sigma} Z.\neg f[\neg Z/Z],$$

one can equivalently convert a formula into its *negation normal form* (NNF). Formulas in NNF can then be described by the following grammar:

$$f ::= \top \mid \bot \mid a \mid \neg a \mid Z \mid \neg Z \mid f \vee g \mid f \wedge g$$
$$\mid \mathsf{X}^{>p} f \mid \mathsf{X}^{\geqslant p} f \mid \mu Z.f \mid \nu Z.f.$$

We denote the variables in f by $\mathcal{Z}(f)$ and the subformulas of f by $\mathcal{S}(f)$. A formula of the form $a, \neg a, \bot$ or $\top$ is called a *literal*.

A formula is said to be *well-named* if each bound variable is bound exactly once, in addition, bound variables and free variables do not share the same names.

For a well-named formula f and a bound variable $Z \in \mathcal{Z}(f)$, the *binder* of Z in f is the subformula $\sigma Z.g \in \mathcal{S}(f)$, denoted by $\mathscr{D}_f(Z)$.

A formula f is *guarded*, if for each bound variable $Z \in \mathcal{Z}(f)$, each of its occurrences must be in the scope of some X operator within $\mathscr{D}_f(Z)$.

Given a well-named formula f, we can define the *dependency relation* $\sqsubset_f$ over its bound variables, this relation is the minimal strict partial order such that Z_1 being free in $\mathscr{D}_f(Z_2)$ implies $Z_1 \sqsubset_f Z_2$.

Example 1 Let

$$f = \mu Z_1.(a_1 \wedge \nu Z_2.(a_2 \vee \mathsf{X}^{>0.7} Z_2 \vee \mathsf{X}^{\geqslant 0.4} Z_1)).$$

Then we have $\mathscr{D}_f(Z_1) = f$ and

$$\mathscr{D}_f(Z_2) = \nu Z_2.(a_2 \vee \mathsf{X}^{>0.7} Z_2 \vee \mathsf{X}^{\geqslant 0.4} Z_1).$$

Z_1 has a free occurrence in $\mathscr{D}_f(Z_2)$, and thus we have $Z_1 \sqsubset_f Z_2$. The formula $\mu Z.(a \wedge \mathsf{X}^{>0.5} Z)$ is guarded, while $\mathsf{X}^{>0.6} \nu Z.(\neg a \wedge Z \vee \mathsf{X}^{\geqslant 0.4} a)$ is not, because the occurrence of Z in its binder is not in the scope of any X operator.

For every $g \in \mathcal{S}(f)$, the *expansion* of g with respect to $\mathscr{D}_f$ is defined as

$$\langle\!| g |\!\rangle_{\mathscr{D}_f} = g[\mathscr{D}_f(Z_n)/Z_n] \cdots [\mathscr{D}_f(Z_1)/Z_1],$$

where the linear order $(Z_1, Z_2, \ldots, Z_n)$ of the bound variables in $\mathcal{Z}(f)$ is compatible with $\sqsubset_f$, i.e., $Z_i \sqsubset_f Z_j$ implies $i < j$. We call the set

$$\{\langle\!| g |\!\rangle_{\mathscr{D}_f} \mid g \in \mathcal{S}(f)\},$$

to be the *Fisher–Ladner closure* of f.

Note that the expansion of g may not be well-named in the case that some bound variable Z appears freely more than once in some $\sigma Z'.g_{Z'}$, but $Z \sqsubset_f Z'$. In this case, a systematic renaming is required.

2.3 Probabilistic alternating parity automata

The notion of *probabilistic alternating parity automata* (PAPA) was first introduced in Ref. [16] to characterize (closed) PμTL formulas.

Recall that a PAPA A is a tuple (Q, q_0, δ, Ω) where

- Q is a finite set of *states*;
- $q_0 \in Q$ is the unique *initial state*;
- δ is the *transition function*, that assigns each state a transition condition over Q (we will define the notion of transition conditions later);
- $\Omega : Q \rightsquigarrow \mathbb{N}$, is a partial function, that assigns an integer to some states.

The set of *transition conditions* over Q can be inductively defined as follows:

(1) Each literal is a transition condition over Q.

(2) Each state $q \in Q$ is a transition condition over Q.

(3) If $q \in Q$ and p is a probability, then $\mathsf{X}^{\gtrsim p} q$ is a transition condition over Q, where $\gtrsim \in \{\geqslant, >\}$.

(4) If q_1 and q_2 are two states in Q, then both $q_1 \vee q_2$ and $q_1 \wedge q_2$ are transition conditions over Q.

Given a pointed Markov chain (M, s_0) where $M = (S, P, L)$ and $s_0 \in S$, then a *run* of A over (M, s_0) is a $Q \times S$-labeled tree (T, λ) that fulfills: $\lambda(v_0) = (q_0, s_0)$, where v_0 is the root vertex; and for each internal vertex v with $\lambda(v) = (q, s)$, we have the following requirements:

- If $\delta(q) = \bot$ or $\delta(q) = \top$ then v has no child.
- If $\delta(q) = a$ then $a \in L(s)$; and if $\delta(q) = \neg a$ then $a \notin L(s)$.
- If $\delta(q) = q'$ then v has one child v' with $\lambda(v') = (q', s)$.
- If $\delta(q) = q_1 \wedge q_2$ then v has two children v_1 and v_2 such that $\lambda(v_1) = (q_1, s)$ and $\lambda(v_2) = (q_2, s)$, respectively.
- If $\delta(q) = q_1 \vee q_2$ then v has one child v', either $\lambda(v') = (q_1, s)$ or $\lambda(v') = (q_2, s)$.
- If $\delta(q) = \mathsf{X}^{\gtrsim p} q'$ then v has a set of children $v_1, \ldots, v_n$ such that $\lambda(v_i) = (q', s_i)$, where $\sum_{i=1}^{n} P(s, s_i) \gtrsim p$.

For an infinite path $\tau = v_0, v_1, \ldots$ of T, we let n_τ be the number

$$\max\{\, n \mid \#\{i \mid \Omega(\mathrm{Proj}_1(\lambda(v_i))) = n\} = \infty\},$$

where $\mathrm{Proj}_1(q, s) = q$ for every $q \in Q$ and $s \in S$.

A run (T, λ) is *accepting* if: (1) n_τ is an even number for every infinite branch τ of T; and (2) for each vertex v of T with $\lambda(v) = (q, s)$, we have $\delta(q) \neq \bot$.

A pointed Markov chain (M, s_0) can be *accepted* by A, if A has some accepting run over it. Let $\mathscr{L}(A)$ be the set of pointed Markov chains that can be accepted by A.

For each closed PμTL formula f, we can create a PAPA $A_f = (Q_f, q_f, \delta_f, \Omega_f)$, where

- $Q_f = \{q_g \mid g \in \mathcal{S}(f)\}$, hence we have the state $q_f \in Q_f$;
- δ_f is defined as follows:
  - $\delta_f(q_\bot) = \bot$ and $\delta_f(q_\top) = \top$;
  - $\delta_f(q_a) = a$ and $\delta_f(q_{\neg a}) = \neg a$;
  - $\delta_f(q_{g_1 \wedge g_2}) = q_{g_1} \wedge q_{g_2}$ and $\delta_f(q_{g_1 \vee g_2}) = q_{g_1} \vee q_{g_2}$;
  - $\delta_f(q_{\mathsf{X}^{\gtrsim p} g}) = \mathsf{X}^{\gtrsim p} q_g$;
  - $\delta_f(q_{\sigma Z.g}) = q_g$;
  - $\delta_f(q_Z) = q_g$ if $\mathscr{D}_f(z) = \sigma Z.g$.
- Ω_f is defined at every state q_Z with $Z \in \mathcal{Z}$ fulfilling: (1) if Z is a μ-variable (resp. ν-variable), then $\Omega_f(q_Z)$ is an odd (resp. even) number; (2) if $Z_1 \sqsubset_f Z_2$, then we require that $\Omega_f(q_{Z_1}) \geqslant \Omega_f(q_{Z_2})$.

Theorem 1 (Ref. [16]) $M, s \models f$ if and only if $(M, s) \in \mathscr{L}(A_f)$.

3 Tableaux

We define the tableaux of PμTL formulas in this section. For convenience, we fix a formula f, and assume that it is (1) closed, (2) well-named, (3) in NNF, and (4) guarded. A formula satisfying these conditions is in *tableau normal-form*. We will later see that under our presented axiom system, every PμTL formula is equivalent to a formula in tableau normal-form.

A set Γ of formulas of the form

$$\{\mathsf{X}^{\gtrsim p_1} f_1, \ldots, \mathsf{X}^{\gtrsim p_n} f_n, l_1, \ldots, l_s\},$$

where $l_1, \ldots, l_s$ are literals, is called a *modal set*. Furthermore, we define $\mathsf{Post}(\Gamma) = \{f_1, \ldots, f_n\}$. A modal set is called (*locally*) *consistent* if and only if it contains neither $\bot$ nor conflicting literals (e.g., a and $\neg a$).

For a modal set $\Gamma \subseteq \mathcal{S}(f)$, a *cover* $\mathcal{C}$ of Γ is a subset of $2^{\mathsf{Post}(\Gamma)}$ such that there exists a weight function $w: \mathcal{C} \to [0, 1]$ satisfying

(1) $\sum_{\Delta \in \mathcal{C}} w(\Delta) \leqslant 1$, and

(2) $\sum_{\Delta \in \{\Delta' \in \mathcal{C} | g \in \Delta'\}} w(\Delta) \gtrsim p$ for every $\mathsf{X}^{\gtrsim p} g \in \Gamma$.

We denote the set of covers of Γ by $\mathscr{C}(\Gamma)$.

To prepare for tableaux, we first list the tableau rules for PμTL.

$$[\text{AND}]\ \frac{\Gamma, f_1, f_2}{\Gamma, f_1 \wedge f_2}; \quad [\text{OR}]\ \frac{\Gamma, f_i}{\Gamma, f_1 \vee f_2},\ i \in \{1, 2\};$$

$$[\mu]\ \frac{\Gamma, g}{\Gamma, \mu Z.g}; \quad [\nu]\ \frac{\Gamma, g}{\Gamma, \nu Z.g};$$

$$[\text{REG}]\ \frac{\Gamma, g}{\Gamma, Z}, \quad \text{if } \mathscr{D}_f(Z) = \sigma Z.g;$$

$$[\text{MOD}]\ \frac{\Gamma_1, \ldots, \Gamma_n}{\Gamma}, \quad \begin{array}{l}\Gamma \text{ is a consistent modal set,}\\ \{\Gamma_1, \ldots, \Gamma_n\} \text{ is a cover of } \Gamma.\end{array}$$

For convenience, we always consider f to be a closed formula in this section. As a matter of fact, free variables can be treated in the same way as literals, and thus this condition is never an actual restriction.

A *tableau* of f is a labeled tree $T_f = (V, v_0, E, L)$, where V denotes the nodes and $v_0 \in V$ denotes the root, E are the edges from parent to child, and $L: V \to 2^{\mathcal{S}(f)}$ labels each node with a set of subformulas of f. Intuitively, the tableau has the following properties: (1) if the labels of the children are satisfiable, then so does the parent's label; (2) the labels of leaf nodes are satisfiable if they are consistent.

We require that $L(v_0) = \{f\}$. And, for each node $v \in V$, if $L(v)$ contains a pair of conflicting literals or $\bot$, then v is a leaf node of the labeled tree; otherwise,

- if $L(v)$ is not a modal set, then v has a single child v' with a label $L(v')$ such that $\dfrac{L(v')}{L(v)}$ is some tableau rule;
- if $L(v)$ is a modal set, then we apply the [MOD]-rule in the following way: we find some cover $\mathcal{C} = \{\Gamma_1, \ldots, \Gamma_n\}$ of $L(v)$ and create n children of v, each is labeled with one distinct element of $\mathcal{C}$ (note that v becomes a leaf if $L(v)$ consists of literals only).

For each edge $(v, v') \in E$, we call v an [X]-*node* if $L(v')$ is obtained from $L(v)$ by applying the [X]-rule. It is indeed well-defined because only [MOD]-nodes may have more than one child.

In addition, if v is neither a [MOD]-node nor an [AND]-node, then $L(v)$ and $L(v')$ must be of the forms Γ, g and Γ, g', respectively. In this case, we say that (g, g') is the *reduction pair* w.r.t. v and v'. If v is an [AND]-node, then we have $L(v) = \Gamma, g_1 \wedge g_2$, and $L(v') = \Gamma, g_i$ where $i \in \{1, 2\}$. Hence we have two reduction pairs $(g_1 \wedge g_2, g_i)$. Otherwise, if v is a [MOD]-node and v' is one of its children, then a reduction pair must be of the form $(\mathsf{X}^{\gtrsim p} g, g)$ where $\mathsf{X}^{\gtrsim p} g \in L(v)$ and $g \in L(v')$.

For the tableau $T_f = (V, v_0, E, L)$, a *(maximum) path* Π is a sequence

$$L(v_0), L(v_1), \ldots, L(v_i), \ldots,$$

where v_0 is the root and each $(v_i, v_{i+1}) \in E$. For convenience, we sometimes call

$$v_0, v_1, \ldots, v_i, \ldots$$

a path, if doing so does not result in ambiguity.

A *trace* π within Π is a sequence of formulas

$$g_0, g_1, \ldots, g_i, \ldots,$$

where $g_i \in L(v_i)$, and each (g_i, g_{i+1}) is (1) either a reduction pair w.r.t. v_i and v_{i+1}; or (2) $g_i = g_{i+1}$ (i.e., g_i is not rewritten in $L(v_i)$).

For some variable Z, if we have $g_i = Z$ and $g_{i+1} = g$, where $\mathscr{D}_f(Z) = \sigma Z.g$, then we say that Z *regenerates* at v_i. In other words, v_i must be a [REG]-node. For an infinite trace, there must exist at least one bound variable that regenerates infinitely often. The reason is that (1) formulas in a trace cannot remain unchanged forever, because a path will eventually encounter [MOD]-nodes; and (2) [REG] is the only tableau rule that can increase formula size.

Given an infinite trace π, we call it a μ-*trace* (resp. ν-trace) if the least variable (w.r.t. $\sqsubset_f$) regenerating infinitely often along it is a μ-variable (resp. ν-variable).

Then, the tableau $T_f = (V, v_0, E, L)$ is said to be *consistent* if it is

locally consistent: $L(v)$ contains no conflicting pair of literals or $\bot$, for each node $v \in V$; and

globally consistent: no μ-trace exists in T_f.

Theorem 2 If f has a consistent tableau, then it is satisfiable.

Proof Let $T_f = (V, v_0, E, L)$ be a tableau. We first construct a Markov chain $M = (S, P, L')$ from T_f in the following way:

(1) For each node $v \in V$, let $\Phi(v)$ be the nearest [MOD] descendant (if v is a [MOD]-node, then just let $\Phi(v) = v$). Note that nodes other than [MOD]-nodes must have a unique child, the function Φ is a well-defined one.

(2) For each [MOD]-node $v \in V$, we create a state s_v in the Markov chain. Let S consist of all such states, in addition to one extra state s_{sink}.

(3) For each $s_v \in S \setminus \{s_{\text{sink}}\}$, we choose $L'(s_v)$ such that it satisfies the following: for each $a \in \mathcal{A}$, if $a \in L(v)$ then $a \in L'(s_v)$; and if $\neg a \in L(v)$ then $a \notin L'(s_v)$. As T_f is locally consistent, $L'(s_v)$ does exist. $L'(s_{\text{sink}})$ can be any subset of $\mathcal{A}$.

(4) The distribution function P is determined as follows. For each [MOD]-node v, if $v_1, v_2, \ldots, v_n$ are all its children, then $\mathcal{C} = \{ L(v_i) \mid i = 1, \ldots, n \}$ forms a cover of $L(v)$. By definition, there exists a weight function w_v from $\mathcal{C}$ to $[0,1]$. Then, we set $P(s_v, s_{\Phi(v_i)}) = w_v(L(v_i))$ for each i, and set $P(s_v, s_{\text{sink}}) = 1 - \sum_{i=1}^{n} w_v(L(v_i))$.

Let A_f be the PAPA of f. To show $(M, s_{\Phi(v_0)})$ is accepted by A_f, we construct an accepting run of A_f on it with the following steps.

(1) For each node $v \in V$ and each $g \in L(v)$, we create a tuple $(g, v, s_\Phi(v))$, all such tuples constitute the node set of the run tree.

(2) We let $(g, v, s_{\Phi(v')})$ be the parent of $(g', v', s_{\Phi(v')})$ if the following hold:

• There exists some trace $\pi = g_0, g_1, \ldots, g_i, \ldots$ within a maximal path $\Pi = L(v_0), L(v_1), \ldots, L(v_i), \ldots$, where each $g_i \in L(v_i)$.

• There exist some $i \leqslant k < j$ such that $v = v_i$ and $v' = v_j$; and $g = g_i = \cdots = g_k$, and $g' = g_{k+1} = \cdots = g_j$, and $g \neq g'$.

(3) For each tuple $(g, v, s_{\Phi(v)})$, we label it with $(q_g, s_{\Phi(v)})$.

Then, according to the construction of A_f and M, one can directly check that this run is indeed accepting because T_f is also globally consistent.

Finally, we have $\mathscr{L}(A_f) \neq \varnothing$, and this implies that f is satisfiable by Theorem 1. ■

The dual concept of tableau is *refutation*. The rules of a refutation are almost the same as those for a tableau, but [OR] and [MOD] are respectively replaced by the following [OR′]- and [MOD′]-rules:

$$[\text{OR}'] \ \frac{\Gamma, f_1 \quad \Gamma, f_2}{\Gamma, f_1 \vee f_2};$$

$$[\text{MOD}'] \ \frac{\Gamma_1, \ldots, \Gamma_m}{\Gamma}, \quad \begin{array}{l}\text{for each } \mathcal{C} \in \mathscr{C}(\Gamma),\\ \text{there is some } \Gamma_i \in \mathcal{C}.\end{array}$$

Therefore, an [OR′]-node labeled with $\Gamma \cup \{f_1 \vee f_2\}$ has two children that are labeled with $\Gamma \cup \{f_1\}$ and $\Gamma \cup \{f_2\}$. Meanwhile, for a [MOD′]-node labeled with Γ, for each $\mathcal{C} \in \mathscr{C}(\Gamma)$, there is a child labeled with some $\Gamma_i \in \mathcal{C}$.

A refutation is *feasible* if (1) either some node has a label containing a conflicting pair of literals or $\bot$; or (2) each infinite path contains a μ-trace.

Lemma 1 For a formula f, it has either a consistent tableau or a feasible refutation.

Proof We define a two-player game (played by Satisfier and Refuter) as follows: For each set $\Gamma \subseteq \mathcal{S}(f)$:

(1) If it is not a modal set, then Satisfier picks some successor Γ' according to some tableau rule other than [MOD].

(2) If it is a modal set, then first Satisfier provides a cover $\mathcal{C}$ of Γ, and then Refuter chooses some $\Gamma' \in \mathcal{C}$ as the successor of Γ.

The game starts from $\{f\}$, and Refuter wins only if

• the path (defined exactly as that in tableaux) has a node that involves either $\bot$ or some conflicting pairs; or

• the path is infinite and contains some μ-trace.

This game is definitely deterministic (i.e., well partitioned and the winning conditions of players are complementary), then according to Martin's theory[30], the game has exactly one winner.

If Satisfier wins, she can construct a consistent tableau of f, and refer to her winning strategy to make suitable choices in [OR]-nodes and [MOD]-nodes. Likewise, if Refuter has a winning strategy, she can establish a feasible refutation by querying her winning strategy in choosing [MOD′]-nodes. ■

Theorem 3 If f is not satisfiable then it has a feasible refutation.

Proof Suppose that f is not satisfiable. From Theorem 2, Satisfier (cf. Lemma 1) cannot have a winning strategy, and Refuter wins the corresponding game, thus, f has a feasible refutation. ■

4 Axiom System

In this section, we extend Kozen's system to that for PμTL as follows:

(TAUT)	All tautologies,
(PROB)	$\mathsf{X}^{>b} f \to \bot$,
(REC)	$f[\mu Z.f/Z] \to \mu Z.f$,

where in (PROB), we require $b \geqslant 1$. In addition, the system includes the inference rules

$$\text{(MP)} \qquad \frac{f \to g, \qquad f}{g},$$

$$\text{(LFP)} \qquad \frac{f[g/Z] \to g}{(\mu Z.f) \to g},$$

$$\text{(COV)} \qquad \frac{\bigvee_{B\in\mathcal{B}} \bigwedge B \to \bot}{\bigwedge \Gamma \to \bot}.$$

For the rule (COV), we require Γ to be a modal set, and $\mathcal{B} \subseteq 2^{\mathsf{Post}(\Gamma)}$ such that $\mathcal{B} \cap \mathcal{C} \neq \varnothing$ for each $\mathcal{C} \in \mathscr{C}(\Gamma)$.

Formally, a *proof* of f is a sequence

$$f_0, f_1, \ldots, f_n = f,$$

where each f_i is either an instance of some axiom or obtained by applying some inference rule from $f_0, \ldots, f_{i-1}$.

We say that f is *provable*, denoted as $\vdash f$, if there exists a proof for f; and, f is *irrefutable* if $\neg f$ is not provable.

Different from Kozen's axiom system, we add the axiom (PROB) and the inference rule (COV). For a set of X-formulas Γ, the probabilities of mutually exclusive formulas in $\mathsf{Post}(\Gamma)$ may be so high that no cover of Γ exists. Then, a complete proof system should allow inference of $\bigwedge \Gamma \to \bot$. The antecedents of (COV) describe the mutually exclusive formulas: every $B \subseteq \mathsf{Post}(\Gamma)$ in $\mathcal{B}$ is such a set of mutually exclusive formulas.

Example 2 Consider the formula set

$$\Gamma = \{\mathsf{X}^{>0.7}a, \mathsf{X}^{>0.7}b, \mathsf{X}^{>0.7}(\neg a \vee \neg b)\},$$

where we definitely have $\mathsf{Post}(\Gamma) = \{a, b, \neg a \vee \neg b\}$.

In addition, we let $\mathcal{B}$ be the set $\{\mathsf{Post}(\Gamma)\}$, the set consists of all maximal proper subsets of $\mathsf{Post}(\Gamma)$, i.e., the set

$$\{\{a,b\}, \{a, \neg a \vee \neg b\}, \{b, \neg a \vee \neg b\}\},$$

is not a cover of Γ, because of the absence of a proper weight function for this set. Therefore, each $\mathcal{C} \in \mathscr{C}(\Gamma)$ must contain $\mathsf{Post}(\Gamma)$, so $\mathcal{B} \cap \mathcal{C} \neq \varnothing$.

Thus we can infer

$$\mathsf{X}^{>0.7}a \wedge \mathsf{X}^{>0.7}b \wedge \mathsf{X}^{>0.7}(\neg a \vee \neg b) \to \bot,$$

from the tautology

$$a \wedge b \wedge (\neg a \vee \neg b) \to \bot,$$

by applying (COV).

4.1 Equivalent formulation of COV

Although the inference rule (COV) appears neat and intuitive, calculating all the covers of a modal set is inconvenient. We thus present an equivalent rule called (COV′), which is relatively easy to use. This rule reads

$$\text{(COV}'\text{)} \qquad \frac{\bigvee_{B\in\mathcal{B}} \bigwedge B \to \bot}{\bigwedge \Gamma \to \mathsf{X}^{\gtrsim p} \bigvee \mathsf{Post}(\Gamma)},$$

where Γ is a modal set, $\mathcal{B} \subseteq 2^{\mathsf{Post}(\Gamma)}$, and p is the optimization goal of the following linear programming (LP) problem:

$$\begin{aligned} p = \quad & \inf \sum_{\Delta:\Delta\subseteq\mathsf{Post}(\Gamma)} w_\Delta, & \\ \text{s.t.} \quad & w_\Delta \geqslant 0, & \forall \Delta \subseteq \mathsf{Post}(\Gamma); \\ & w_\Delta = 0, & \forall \Delta \in \mathcal{B}; \\ & \sum_{\Delta:g\in\Delta} w_\Delta \gtrsim p', & \forall \mathsf{X}^{\gtrsim p'} g \in \Gamma; \end{aligned}$$

where

$$\gtrsim \; = \begin{cases} \geqslant, & \text{if the infimum is achievable;} \\ >, & \text{otherwise.} \end{cases}$$

Note that (COV′) can be applied only when this LP problem has a solution.

Example 3 Consider the formula set

$$\Gamma = \{\mathsf{X}^{>0.7}a, \mathsf{X}^{>0.7}b, \mathsf{X}^{>0.7}(\neg a \vee \neg b)\},$$

and still let $\mathcal{B} = \{\mathsf{Post}(\Gamma)\}$. Then, we can obtain the following LP problem:

$$\begin{aligned} & p = \inf \textstyle\sum_{\Delta:\Delta\subseteq\mathsf{Post}(\Gamma)} w_\Delta, \quad \text{s.t.} \\ & w_\Delta \geqslant 0, \quad \forall \Delta \subseteq \mathsf{Post}(\Gamma); \\ & w_{\{a,b,\neg a\vee\neg b\}} = 0, \\ & w_{\{a\}} + w_{\{a,b\}} + w_{\{a,\neg a\vee\neg b\}} + w_{\{a,b,\neg a\vee\neg b\}} > 0.7, \\ & w_{\{b\}} + w_{\{a,b\}} + w_{\{b,\neg a\vee\neg b\}} + w_{\{a,b,\neg a\vee\neg b\}} > 0.7, \\ & w_{\{\neg a\vee\neg b\}} + w_{\{a,\neg a\vee\neg b\}} + w_{\{b,\neg a\vee\neg b\}} + \\ & \quad w_{\{a,b,\neg a\vee\neg b\}} > 0.7. \end{aligned}$$

One may check that its minimum value $p = 1.05$ is not achievable, we thus have

$$\mathsf{X}^{>0.7}a \wedge \mathsf{X}^{>0.7}b \wedge \mathsf{X}^{>0.7}(\neg a \vee \neg b) \to \bot,$$

by applying (COV′) to

$$a \wedge b \wedge (\neg a \vee \neg b) \to \bot.$$

Note that we have

$$\mathsf{X}^{>1.05}(a \vee b \vee \neg a \vee \neg b) \to \bot,$$

according to (PROB).

Let $\mathscr{A} = \{\text{(TAUT)}, \text{(PROB)}, \text{(REC)}, \text{(MP)}, \text{(LFP)}\}$. Then, Theorems 4 and 5 prove the equivalence of (COV) and (COV′) cooperating with $\mathscr{A}$.

Theorem 4 The system $\mathscr{A} \cup \{\text{(COV)}\}$ can derive (COV′).

Proof Suppose Γ is a modal set, and let $\mathcal{B} \subseteq 2^{\mathsf{Post}(\Gamma)}$ be such a set family that for every $B \in \mathcal{B}$ we can infer $\bigwedge B \to \bot$. In this situation, the premise of (COV′) is satisfied. Suppose that $\gtrsim p$ is determined by the corresponding LP problem. At this point, the obligation is to show

$$\bigwedge \Gamma \to \mathsf{X}^{\gtrsim p} \bigvee \mathsf{Post}(\Gamma).$$

For convenience, we prove the following equivalent formulation:

$$\bigwedge \Gamma \wedge \Big(\mathsf{X}^{\gtrapprox 1-p} \bigwedge_{g\in\mathsf{Post}(\Gamma)} \neg g\Big) \to \bot \qquad (1)$$

remind that $\neg\mathsf{X}^{\gtrapprox 1-p}\neg g$ is equivalent to $\mathsf{X}^{\gtrsim p} g$ in our logic, and $\gtrapprox \in \{>, \geqslant\} \setminus \{\gtrsim\}$.

To prepare the premises of (Cov); and we let

$$\Gamma' = \Gamma \cup \{\mathsf{X}^{\gtrapprox 1-q} h\},$$

where $h = \bigwedge_{g\in\mathsf{Post}(\Gamma)} \neg g$ and let

$$\mathcal{B}' = \mathcal{B} \cup \{\, \Delta \subseteq \mathsf{Post}(\Gamma') \mid h \in \Delta \,\} \setminus \{\{h\}\}.$$

Note that each $\Delta \in \mathcal{B}'\backslash\mathcal{B}$ contains some $g \in \mathsf{Post}(\Gamma)$ and $h = \bigwedge_{g\in\mathsf{Post}(\Gamma)} \neg g$. Therefore, we have $\bigwedge \Delta \to \bot$ for each $\Delta \in \mathcal{B}'$.

In addition, for every cover $\mathcal{C}$ of Γ' there must exist some weight function $w: \mathcal{C} \to [0, 1]$ fulfilling

$$\sum_{\Delta\in\mathcal{C}} w(\Delta) \leqslant 1 \qquad (2)$$

$$\sum_{\Delta\in\{\,\Delta'\in\mathcal{C}\mid h\in\Delta'\,\}} w(\Delta) \gtrapprox 1-p \qquad (3)$$

$$\sum_{\Delta\in\{\,\Delta'\in\mathcal{C}\mid g\in\Delta'\,\}} w(\Delta) \gtrsim q \quad \forall \mathsf{X}^{\gtrsim q} g \in \Gamma \qquad (4)$$

We can now define the extension $w': 2^{\mathsf{Post}(\Gamma')} \to [0, 1]$ as follows:

$$w'(\Delta) = \begin{cases} w(\Delta), & \text{if } \Delta \in \mathcal{C}; \\ 0, & \text{otherwise.} \end{cases}$$

Then the above can be reformulated as

$$\sum_{\Delta\in 2^{\mathsf{Post}(\Gamma')}} w'(\Delta) \leqslant 1 \qquad (5)$$

$$\sum_{\Delta\in\{\Delta'\in 2^{\mathsf{Post}(\Gamma')}\mid\, h\in\Delta'\}} w'(\Delta) \gtrapprox 1-p \qquad (6)$$

$$\sum_{\Delta\in\{\,\Delta'\in 2^{\mathsf{Post}(\Gamma')}\mid\, g\in\Delta'\,\}} w'(\Delta) \gtrsim q \quad \forall \mathsf{X}^{\gtrsim q} g \in \Gamma \qquad (7)$$

Assume by contradiction that $\mathcal{C} \cap \mathcal{B}' = \varnothing$, then we have the following observations:

- The summation in Formula (7) accounts for all such Δ that either it is a subset of $2^{\mathsf{Post}(\Gamma)}$ or it contains h. For the second case, it does not belong to $\mathcal{C}$.
- According to the definition of w', we have $w'(\Delta) = 0$. Thus, for every $\mathsf{X}^{\gtrsim q} g \in \Gamma$, we can assert that

$$\sum_{\Delta\in\{\,\Delta'\in 2^{\mathsf{Post}(\Gamma)}\mid g\in\Delta\,\}} w'(\Delta') \gtrsim q.$$

- The restriction of w' to $2^{\mathsf{Post}(\Gamma)}$ also satisfies the first two requirements of the corresponding LP problem, hence we have

$$\sum_{\Delta\in 2^{\mathsf{Post}(\Gamma)}} w'(\Delta) \gtrsim p.$$

- From Formula (6) we have

$$\sum_{\Delta\in 2^{\mathsf{Post}(\Gamma)}} w'(\Delta) \not\gtrapprox p,$$

thus we have a contradiction (take note of the difference between $\gtrapprox$ and $\not\gtrapprox$).

Therefore, the assumption $\mathcal{C} \cap \mathcal{B}' = \varnothing$ is incorrect. Now, we can apply (Cov) with Γ' and $\mathcal{B}'$ and then can obtain Formula (1). ■

Theorem 5 The system $\mathscr{A} \cup \{(\text{Cov}')\}$ can derive (Cov).

Proof Suppose that we have a modal set $\Gamma \subseteq \mathcal{S}(f)$ and a set $\mathcal{B} \subseteq 2^{\mathsf{Post}(\Gamma)}$ such that for every $\mathcal{C} \in \mathscr{C}(\Gamma)$, we have $\mathcal{C} \cap \mathcal{B} \neq \varnothing$. As we have the axiom (Prob), it suffices to show that the optimal value of the LP problem is greater than or equal to 1, but this value is not achievable.

Assume by contradiction that the optimal value is less than 1 (or, equals to 1 and it is achievable), then there must exist a solution having $\sum_{\Delta\in 2^{\mathsf{Post}(\Gamma)}} w_\Delta \leqslant 1$. As a result, the set

$$\mathcal{C} = \{\, \Delta \in 2^{\mathsf{Post}(\Gamma)} \mid w_\Delta > 0 \,\},$$

is a cover of Γ. By assumption, we have $\mathcal{C} \cap \mathcal{B} \neq \varnothing$. This contradicts the fact that $w_\Delta = 0$ for every $\Delta \in \mathcal{B}$. ■

4.2 Some provable formulas

In this section, we present some useful formulas that are provable in the presented PμTL axiom system.

As our axiom system subsumes Kozen's system, we can show the following theorems analogously by adapting the quantifiers with the probabilistic version.

Theorem 6 For each closed formula f there exists some g in tableau normal form such that $\vdash f \leftrightarrow g$.

The proof of the fact that every formula can be equivalently transformed into a positive guarded one under Kozen's axiom system can be found in Ref. [2, Proposition 2].

Theorem 7 (Ref. [1]) Given a variable Z and three formulas f, g, and h, suppose that all occurrences of Z in f and g are positive, and let $\sigma \in \{\mu, \nu\}$. Then,

(1) If $\vdash f \to g$ then $\vdash f[h/Z] \to g[h/Z]$.
(2) If $\vdash f_1 \to f_2$ then $\vdash g[f_1/Z] \to g[f_2/Z]$.
(3) If $\vdash f \to g$ then $\vdash \sigma Z.f \to \sigma Z.g$.
(4) $\vdash \sigma Z.f \leftrightarrow f[\sigma Z.f/Z]$.
(5) If $\vdash f[\mu Z.(g \wedge f)/Z] \to g$ then $\vdash \mu Z.f \to g$.

(6) Supposing $Z_1, \ldots, Z_n$ are free variables of f, and $a_1, \ldots, a_n \notin \mathcal{L}(f)$, then $\vdash f$ if and only if $\vdash f[a_1/Z_1]\ldots[a_n/Z_n]$.

We now proceed to some formulas characterizing probabilistic quantifiers. Their proofs demonstrate how (PROB) and (COV) work.

Theorem 8 With the axiom system, we can show the following:

(1) $\vdash \mathsf{X}^{>p} f \wedge \mathsf{X}^{>p'} g \to \mathsf{X}^{>p+p'-1}(f \wedge g)$.

(2) $\vdash \mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{>p-p'}(f \wedge \neg g)$.

(3) $\vdash \neg\mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{\geqslant 1-p-p'}(\neg f \wedge \neg g)$.

(4) $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{\geqslant p} f$.

(5) $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{>p-p'} f$.

(6) $\vdash \mathsf{X}^{\geqslant p} f \to \mathsf{X}^{\geqslant p-p'} f$.

(7) $\vdash \mathsf{X}^{\geqslant p} f \to \mathsf{X}^{>p-p'} f$ where $p' > 0$.

(8) If $\vdash f \to \bot$ then $\vdash \mathsf{X}^{\geqslant p} f \to \bot$.

(9) If $\vdash f$ then $\vdash \mathsf{X}^{\geqslant 1} f$.

(10) If $\vdash f \to g$ then $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{>p} g$.

(11) If $\vdash f \to g$ then $\vdash \mathsf{X}^{\geqslant p} f \to \mathsf{X}^{\geqslant p} g$.

Proof First of all, we can see that the hypothetical syllogism (HS) can be proven: i.e., if $\vdash f \to g$ and $\vdash g \to h$ then we have $\vdash f \to h$. As

$$(f \to g) \to ((g \to h) \to (f \to h)),$$

is a tautology, we thus derive the conclusion by applying (MP) twice.

(1) $\vdash \mathsf{X}^{>p} f \wedge \mathsf{X}^{>p'} g \to \mathsf{X}^{>p+p'-1}(f \wedge g)$.

(a) $f \wedge g \wedge \neg(f \wedge g) \to \bot$, (TAUT);

(b) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{>p'} g \wedge \mathsf{X}^{\geqslant 1-(p+p'-1)} \neg(f \wedge g) \to \mathsf{X}^{>1}(f \vee g \vee \neg(f \wedge g))$, (a) and (COV′);

(c) $\mathsf{X}^{>1}(f \vee g \vee \neg(f \wedge g)) \to \bot$, (PROB);

(d) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{>p'} g \wedge \mathsf{X}^{\geqslant 1-(p+p'-1)} \neg(f \wedge g) \to \bot$, (b), (c), and (HS);

(e) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{>p'} g \to \mathsf{X}^{>p+p'-1}(f \wedge g)$, (d), (TAUT), and (MP).

(2) $\vdash \mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{>p-p'}(f \wedge \neg g)$.

(a) $f \wedge \neg g \wedge \neg(f \wedge \neg g) \to \bot$, (TAUT);

(b) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{\geqslant 1-p'} \neg g \wedge \mathsf{X}^{\geqslant 1-(p-p')} \neg(f \wedge \neg g) \to \mathsf{X}^{>1}(f \vee \neg g \vee \neg(f \wedge \neg g))$, (a) and (COV′);

(c) $\mathsf{X}^{>1}(f \vee \neg g \vee \neg(f \wedge \neg g)) \to \bot$, (PROB);

(d) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{\geqslant 1-p'} \neg g \wedge \mathsf{X}^{\geqslant 1-(p-p')} \neg(f \wedge \neg g) \to \bot$, (b), (c), and (HS);

(e) $\mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{>p-p'}(f \wedge \neg g)$, (d), (TAUT), and (MP).

(3) $\vdash \neg\mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{\geqslant 1-p-p'}(\neg f \wedge \neg g)$.

(a) $\neg f \wedge \neg g \wedge \neg(\neg f \wedge \neg g) \to \bot$, (TAUT);

(b) $\mathsf{X}^{\geqslant 1-p} \neg f \wedge \mathsf{X}^{\geqslant 1-p'} \neg g \wedge \mathsf{X}^{>1-(1-p-p')} \neg(\neg f \wedge \neg g) \to \mathsf{X}^{>1} \neg f \vee \neg g \vee \neg(\neg f \wedge \neg g)$, (a) and (COV′);

(c) $\mathsf{X}^{>1} \neg f \vee \neg g \vee \neg(\neg f \wedge \neg g) \to \bot$, (PROB);

(d) $\mathsf{X}^{\geqslant 1-p} \neg f \wedge \mathsf{X}^{\geqslant 1-p'} \neg g \wedge \mathsf{X}^{>1-(1-p-p')} \neg(\neg f \wedge \neg g) \to \bot$, (b), (c), and (HS);

(e) $\neg\mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} g \to \mathsf{X}^{\geqslant 1-p-p'}(\neg f \wedge \neg g)$, (d), (TAUT), and (MP).

(4) $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{\geqslant p} f$.

(a) $f \wedge \neg f \to \bot$, (TAUT);

(b) $\mathsf{X}^{>0}(f \wedge \neg f) \to \bot$, (a) and (COV);

(c) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{>1-p} \neg f \to \mathsf{X}^{>0}(f \wedge \neg f)$, Theorem 8 (1);

(d) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{>1-p} \neg f \to \bot$, (b), (c), and (HS);

(e) $\mathsf{X}^{>p} f \to \mathsf{X}^{\geqslant p} f$, (d), (TAUT), and (MP).

(5) $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{>p-p'} f$.

(a) $\bot \to \bot$, (TAUT);

(b) $\mathsf{X}^{>p'} \bot \to \bot$, (PROB);

(c) $\top \to \neg\mathsf{X}^{>p'} \bot$, (TAUT);

(d) $\neg\mathsf{X}^{>p'} \bot$, (b), (c), and (MP);

(e) $\mathsf{X}^{>p} f \wedge \neg\mathsf{X}^{>p'} \bot \to \mathsf{X}^{>p-p'}(f \wedge \neg\bot)$, Theorem 8 (2);

(f) $\mathsf{X}^{>p} f \to \mathsf{X}^{>p-p'} f$, (e), (TAUT), and (MP).

(6) $\vdash \mathsf{X}^{\geqslant p} f \to \mathsf{X}^{\geqslant p-p'} f$.

(a) $\mathsf{X}^{>1-(p-p')} \neg f \to \mathsf{X}^{>1-(p-p')-p'} \neg f$, Theorem 8 (5);

(b) $\neg\mathsf{X}^{>1-p} \neg f \to \neg\mathsf{X}^{>1-(p-p')} \neg f$, (TAUT);

(c) $\mathsf{X}^{\geqslant p} f \to \mathsf{X}^{\geqslant p-p'} f$, (b), (TAUT), and (MP).

(7) $\vdash \mathsf{X}^{\geqslant p} f \to \mathsf{X}^{>p-p'} f$, where $p' > 0$.

(a) $f \wedge \neg f \to \bot$, (TAUT);

(b) $\mathsf{X}^{\geqslant p} f \wedge \mathsf{X}^{\geqslant 1-(p-p')} \neg f \to \mathsf{X}^{>1}(f \vee \neg f)$, (a) and (COV′);

(c) $\mathsf{X}^{>1}(f \vee \neg f) \to \bot$, (PROB);

(d) $\mathsf{X}^{\geqslant 1+p-(p-p')}(f \vee \neg f) \to \bot$, (b), (c), and (HS);

(e) $\mathsf{X}^{\geqslant p} f \to \mathsf{X}^{>p-p'} f$, (d), (TAUT), and (MP).

(8) If $\vdash f \to \bot$ then $\vdash \mathsf{X}^{\geqslant p} f \to \bot$.

It is immediate form (COV′)

(9) If $\vdash f$ then $\vdash \mathsf{X}^{\geqslant 1} f$.

(a) $f \to (\neg f \to \bot)$, (TAUT);

(b) $\neg f \to \bot$, (a), premise, and (MP);

(c) $\mathsf{X}^{>0} \neg f \to \bot$, Theorem 8 (8);

(d) $(\mathsf{X}^{>0} \neg f \to \bot) \to \neg\mathsf{X}^{>0} \neg f$, (TAUT);

(e) $\mathsf{X}^{\geqslant 1} f$, (c), (d), and (MP).

(10) If $\vdash f \to g$ then $\vdash \mathsf{X}^{>p} f \to \mathsf{X}^{>p} g$.

(a) $\mathsf{X}^{\geqslant 1}(f \to g)$, Theorem 8 (7);

(b) $\mathsf{X}^{>p} f \wedge \mathsf{X}^{\geqslant 1}(f \to g) \to \mathsf{X}^{>p} g$, Theorem 8 (2);

(c) $\mathsf{X}^{>p} f \to \mathsf{X}^{>p} g$, (a), (b), and (MP).

(11) If $\vdash f \rightarrow g$ then $\vdash \mathsf{X}^{\geqslant p} f \rightarrow \mathsf{X}^{\geqslant p} g$.

(a) $\mathsf{X}^{\geqslant 1}(f \rightarrow g)$, Theorem 8 (7);

(b) $\mathsf{X}^{\geqslant p} f \wedge \mathsf{X}^{\geqslant 1}(f \rightarrow g) \rightarrow \mathsf{X}^{\geqslant p} g$, Theorem 8 (3);

(c) $\mathsf{X}^{\geqslant p} f \rightarrow \mathsf{X}^{\geqslant p} g$, (a), (b), and (MP). ■

5 Soundness and Completeness

In this section, we discuss the soundness and completeness (for aconjunctive formulas) of the presented axiom system.

Theorem 9 (Soundness) The axiom system is sound, i.e., $\vdash f$ implies $\models f$.

Proof By induction on the proof sequence. Consider the base case. As $\mathsf{X}^{>1} f$ is not satisfiable, we have $\models \mathsf{X}^{>1} f \rightarrow \bot$. The other axioms can be handled similarly as in standard μTL.

For the induction step, we only consider (COV′), as the other cases are also similar to the nonprobabilistic settings.

Let Γ be a modal set and $\mathcal{B} \subseteq 2^{\mathsf{Post}(\Gamma)}$. Suppose that we have $\models \bigwedge B \rightarrow \bot$ for all $B \in \mathcal{B}$ and that $\gtrsim p$ is determined by its corresponding LP problem. Our goal is to prove

$$\models \bigwedge \Gamma \rightarrow \mathsf{X}^{\gtrsim p} \bigvee \mathsf{Post}(\Gamma).$$

It suffices to prove that for every Markov chain M and state s such that $M, s \models \bigwedge \Gamma$, we have

$$\sum_{t: M,t \models \bigvee \mathsf{Post}(\Gamma)} P(s,t) \gtrsim p \qquad (8)$$

We divide the state space into equivalence classes according to which formulas in $\mathsf{Post}(\Gamma)$ a state satisfies. Formally, for each state t, let

$$\Delta_t = \{ g \in \mathsf{Post}(\Gamma) \mid M, t \models g \}.$$

Then, each $S_\Delta = \{ t \mid \Delta_t = \Delta \}$ indeed forms an equivalence class (or is empty). Furthermore, we define $w_\Delta = \sum_{t \in S_\Delta} P(s,t)$ as the probability that a one-step transition from s enters the class S_Δ. Hence, $w_\Delta \geqslant 0$ for each $\Delta \subseteq \mathsf{Post}(\Gamma)$.

The sum on the left-hand side of Formula (8) can be rewritten as

$$\sum_{t: M,t \models \bigvee \mathsf{Post}(\Gamma)} P(s,t) = \sum_{\Delta \neq \varnothing} \sum_{t \in S_\Delta} P(s,t) = \sum_{\Delta \neq \varnothing} w_\Delta.$$

Note that for all $\mathsf{X}^{\gtrsim p} g \in \Gamma$, we have $M, s \models \mathsf{X}^{\gtrsim p} g$. In other words, by summing up per equivalence class,

$$\sum_{t: M,t \models g} P(s,t) = \sum_{\Delta: g \in \Delta} w_\Delta \gtrsim p,$$

holds. In addition, by induction hypothesis, for all $B \in \mathcal{B}$ we have $w_B = 0$. Hence all the constraints of the LP problem are fulfilled. Therefore, $w_\Delta | \Delta \neq \varnothing$ is a valid solution to the LP problem, that is, $\sum_{\Delta \neq \varnothing} w_\Delta \gtrsim p$. ■

For a PμTL formula f, we say a bound variable Z is *active* in $g \in \mathcal{S}(f)$, if either Z is free in g, or another bound variable Z' occurs in g such that $Z \sqsubset_f Z'$.

Let Z be a variable with its natural binder function $\mathscr{D}_f(Z) = \mu Z.g$. The variable Z is called *aconjunctive* if for each subformula of the form $g_1 \wedge g_2$, we require that Z cannot be active in both g_1 and g_2. A formula is *aconjunctive* if all its μ-variables are aconjunctive.

We call a refutation *thin* if and only if no variable is active in both g_1 and g_2 whenever a formula of the form $g_1 \wedge g_2$ is reduced in some node of the refutation. Similar to μTL, since variables cannot be active in both conjuncts of an aconjunctive formula, the refutation of any aconjunctive formula is thin.

For thin refutations, we have the following important theorem.

Theorem 10 For a closed aconjunctive formula, its negation is provable if it has a thin refutation.

Proof This proof is adapted from that of Kozen in Ref. [1] and that of Walukiewicz in Ref. [2] to the probabilistic case. Note that some modifications are crucial, and we will address them in our proof.

Suppose that formula f has a thin refutation of R_f. From Theorem 6, we can also suppose that f is in tableau normal form. The proof idea is to assign some formula g_n to each node n of R_f and show that each g_n is provable. As the root node is assigned $\neg f$, it concludes the proof.

First of all, we need to attach a list of *tokens* to each node. A token t is labeled with a pair (g, Z) where $g \in \mathcal{S}(f)$ and Z is a bound variable of f. Meanwhile, each token is associated with a *counter*, which records the number of regenerations of the bound variable since its creation. Tokens in the list can be removed. Meanwhile, one can add a new token to the right of this list.

Let us recall some notions and operations defined in Ref. [2]. We say that g is *replaceable* by h in some list of tokens if one of the conditions holds:

(1) h does not appear in any label of a token in the current list.

(2) $Z_g \sqsubset_f Z_h$, where

$$Z_g = \min\{Z \mid (g, Z) \text{ appears in the list}\}$$

and

$$Z_h = \{Z \mid (h, Z) \text{ appears in the list}\}.$$

(3) Z_g and Z_h defined above are the same, but the token labeled with (g, Z_g) in the list is to the left of that

labeled with (h, Z_h).

When g is replaceable by h, then we can perform the following (called *replace g by h*): delete all tokens like (h, Z) (for all $Z \in \mathcal{Z}$), and then replace every (g, Z) by (h, Z). Meanwhile, if g is not replaceable by h, we can remove all tokens labeled with (g, Z) for some variable Z.

We assign an empty token list to the root of R_f. Suppose that we have already assigned a token list to an internal node m. Then, with the following rules, we construct the token list for its child node n:

(1) If [OR′] is applied in m to $g_1 \vee g_2$, then m has two children. Suppose that g_i (herein $i \in \{1, 2\}$) appears in the label of n. The token list for n is obtained by replacing $g_1 \vee g_2$ with g_i if $g_1 \vee g_2$ is replaceable by g_i.

(2) Suppose that [REG] is applied in m to variable Z, and that $\mathscr{D}_f(Z) = \sigma Z.g$. We replace Z with g if Z is replaceable by g. In the case that Z is a μ-variable, we also need to increase the counter of the token labeled with (g, Z), and set the counters of the tokens to the right of that to 0.

(3) Suppose that $[\sigma]$ is applied in m to $\sigma Z.g$ (herein σ is either μ or ν). We need to replace $\sigma Z.g$ with g if the former is replaceable by the latter. In addition, if Z is a μ-variable we need to create a new token labeled with (g, Z), add it to the end of the token list, and set its counter to 0.

(4) Suppose [AND] is applied in m to $g_1 \wedge g_2$. As R_f is thin, every μ-variable is active in at most one of g_i (herein $i = 1, 2$). For each token labeled with $(g_1 \wedge g_2, Z)$, we replace $g_1 \wedge g_2$ with g_i wherein Z is active, if $g_1 \wedge g_2$ is replaceable by g_i $(i = 1, 2)$. Note that at this point, the thinness of the refutation is used.

(5) Suppose that [MOD′] is applied in m, and let $L_T(m) = \{\mathsf{X}^{\gtrsim p_1} f_1, \dots, \mathsf{X}^{\gtrsim p_n} f_n, l_1, \dots, l_s\}$ and $L_T(n) \subseteq \mathsf{Post}(L_T(m))$ for each of its child n. For every token labeled with $(\mathsf{X}^{\gtrsim p_i} f_i, Z)$ such that $f_i \in L_T(n)$, we simply need to replace its label by (f_i, Z).

Lastly, we need to remove all tokens that are either labeled with (g, Z) such that Z is not active in g, or labeled with a formula not appearing in what labeled to the node.

In the same way as that in Ref. [2], we can also show that for every path of R_f there exists a token whose counter becomes arbitrarily big.

Subsequently, we need to assign a formula to each node of R_f. Similar to that done in Refs. [1, 2], for every node n of R_f and every formula $g \in L_T(n)$ we define a new version of binder function $\mathscr{D}_{n,g}$. For some μ-variables Z, we have

$$\mathscr{D}_{n,g}(Z) = \mu Z.\neg\gamma_1 \wedge \cdots \wedge \neg\gamma_k \wedge g_Z,$$

where the formulas $\gamma_1, \dots, \gamma_k$ are determined as follows:

- Suppose that n_0 is the nearest ancestor of n whose list includes a token labeled with (g, Z) and that its counter is just 0.
- Along the path from n_0 to n, suppose that $n_1, \dots, n_k$ are all the nodes having the counter of this token increased; then, for $i = 1, \dots, k$, let

$$\gamma_i = \bigwedge\{ \langle\!|h|\!\rangle_{\mathscr{D}_{n_i,h}} \mid h \in L_T(n_i), h \neq g_Z \}.$$

- Since f is aconjunctive, each γ_i is well-defined.

The formula assigned to the node n is just

$$\neg \bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{n,g}} \mid g \in L_T(n) \} \tag{9}$$

For our construction, we have the following proposition.

Proposition 1 For a node m, if its associated formula $\neg \bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{m,g}} \mid g \in L_T(m) \}$ is unprovable, then there is a child n of m such that $\neg \bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{n,g}} \mid g \in L_T(n) \}$ is unprovable.

Proof The proof is done by cases depending on the rule applied to m, for PμTL, the only new case is that for [MOD′].

The [MOD′] case is a deduction of the (COV)-rule. Let m be such [MOD′]-node. According to the rules in constructing tokens, the tokens in any child node n of m are just renamed or deleted. Hence $\langle\!|h|\!\rangle_{\mathscr{D}_{m,h}} \to \langle\!|h|\!\rangle_{\mathscr{D}_{n,h}}$ for all $h \in L_T(n)$. If no child satisfies the property we want, then we have a proof of

$$\bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{m,g}} \mid g \in L_T(n) \} \to \bot$$

for all children n. We set $\Gamma = L_T(m)$ and $\mathcal{B} = \{ L_T(n) \mid n \text{ is a child of } m \}$. Then all the premises of (COV) are fulfilled. We obtain $\vdash \neg \bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{m,g}} \mid g \in L_T(m) \}$, which is a contradiction. Hence there is at least one child such that $\neg \bigwedge\{ \langle\!|g|\!\rangle_{\mathscr{D}_{n,g}} \mid g \in L_T(n) \}$ is unprovable.

Most of the other cases are routine, and the only tough case is the regeneration of a μ-variable where $\mathscr{D}_f(Z) = \mu Z.g$. Luckily this case is essentially the same as the case in the proof for μTL, and the proof details of this case are provided in Ref. [2]. ■

Back to Theorem 10, we have $\mathscr{D}_{r,f} = \mathscr{D}_f$ for the root r of R_f. Assume that $\neg f$ is unprovable, from Proposition 1 we then immediately obtain an infinite path Π of R_f along which every node's associated formula is unprovable.

Note that the following proof is different from that in Ref. [2].

Let t be the token whose counter can be unbounded on this path. Let Z be a bound μ-variable occurring in the label of t, and suppose $\mathscr{D}_f(Z) = \mu Z.g$.

As the counter of t can be arbitrarily large, two nodes $\hat{n}_1$ and $\hat{n}_2$ must exist on Π such that

(1) $\hat{n}_1$ is an ancestor of $\hat{n}_2$, the parent of $\hat{n}_1$ is a [REG]-node, and $\hat{n}_2$ itself is a [REG]-node;

(2) $L_T(\hat{n}_1) = \{g\} \cup \Gamma$ and $L_T(\hat{n}_2) = \{Z\} \cup \Gamma$, where $g \notin \Gamma$ and $Z \notin \Gamma$;

(3) in $\hat{n}_1$ and $\hat{n}_2$, the parts in the token list to the left of t are the same;

(4) the token t is labelled (g, Z) at $\hat{n}_1$, and is labelled (Z, Z) at $\hat{n}_2$;

(5) the counter of t has increased at least once, and has not reset between $\hat{n}_1$ and $\hat{n}_2$.

At this point, we will show that the formula $\neg \bigwedge\{ \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \mid h \in L_T(\hat{n}_2) \}$ must be provable.

According to the definition in Formula (9), from Theorem 7 (1) we have that

$$\vdash \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \rightarrow \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_1,h}} \tag{10}$$

for every formula $h \in L_T(\hat{n}_1)$ if $h \neq Z$, because: (1) all formulas occurring in R are in NNF, and (2) tokens to the right of t have reset their counters at node $\hat{n}_1$, whereas others remain unchanged.

From the assumption, we have

$$\mathscr{D}_{\hat{n}_1,Z}(Z) = \mu Z.\neg\gamma_1 \wedge \cdots \wedge \neg\gamma_i \wedge g,$$
$$\mathscr{D}_{\hat{n}_2,Z}(Z) = \mu Z.\neg\gamma_1 \wedge \cdots \wedge \neg\gamma_j \wedge g,$$

where $j > i$ and $\gamma_1, \ldots, \gamma_j$ are determined by Formula (9). On one hand, we have

$$\gamma_i = \bigwedge\{ \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_1,h}} \mid h \in L_T(\hat{n}_1), h \neq g \} = \bigwedge_{h\in\Gamma} \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_1,h}}.$$

According to Formula (10), we can infer

$$\vdash \bigwedge\{\langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \mid h \in \Gamma\} \rightarrow \gamma_i \tag{11}$$

Thus we have

$$\neg \bigwedge\{ \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,g}} \mid h \in L_T(\hat{n}_2) \} = \neg \left(\bigwedge\{\langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \mid h \in \Gamma\} \wedge \langle\!| Z |\!\rangle_{\mathscr{D}_{\hat{n}_2,Z}}\right) = \neg \left(\bigwedge\{\langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \mid h \in \Gamma\} \wedge \mu Z.(\neg\gamma_1 \wedge \cdots \wedge \neg\gamma_j \wedge g)\right) = \neg \left(\bigwedge\{\langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,h}} \mid h \in \Gamma\} \wedge g'\right) \tag{12}$$

where $g' = \mu Z.(\neg\gamma_1 \wedge \cdots \wedge \neg\gamma_i \wedge \cdots \wedge \neg\gamma_j \wedge g)$. Recall that from Theorem 7 (4), we have

$$g' \leftrightarrow \neg\gamma_1 \wedge \cdots \wedge \neg\gamma_i \wedge \cdots \wedge \neg\gamma_j \wedge g[g'/Z] \tag{13}$$

We can thus conclude that $\neg \bigwedge\{ \langle\!| h |\!\rangle_{\mathscr{D}_{\hat{n}_2,g}} \mid h \in L_T(\hat{n}_2) \}$ must be a tautology from Formulas (11) – (13).

Therefore, we obtain a contradiction with the assumption that $\neg f$ is unprovable. ■

Finally, we have the following complete theorem for aconjunctive formulas.

Theorem 11 (Completeness) For a PμTL formula such that $\neg f$ is aconjunctive, we have that $\models f$ implies $\vdash f$.

Proof Let $Z_1, \ldots, Z_n$ be all free variables occurring in f, let $a_1, \ldots, a_n$ be n new atomic propostions not belonging to $\mathcal{L}(f)$ and let $f' = f[a_1/Z_1] \ldots [a_n/Z_n]$.

As f is valid, so is f', thus the closed formula $\neg f'$ is not satisfiable. According to Theorem 3, we can build a thin refutation of it because $\neg f'$ is also aconjunctive. Therefore, $\neg\neg f'$ is provable according to Theorem 10. Note that we also have $\vdash \neg\neg f' \rightarrow f'$ by (TAUT), and it implies that $\vdash f'$. Finally, we have $\vdash f$ according to Theorem 8 (6). ■

6 Further Discussion

We extend Kozen's and Walukiewicz's axiomatization to that of PμTL. We also show that the presented axiom system is complete for aconjunctive formulas.

In Ref. [2], Walukiewicz showed that Kozen's axiom system is complete for the whole logic of μTL. Let us briefly recall the key idea of his last step.

- Having shown the completeness for aconjunctive formulas, the negation of a valid formula can be rewritten into the aconjunctive form.
- Instead, Walukiewicz[2] managed to show a stronger proposition: for every formula f, some so-called "*disjunctive formula*" g exists, such that $f \rightarrow g$ is provable. Disjunctive formulas enjoy several advantages, in particular that they have very simple tableaux.
- One may perform a proof by induction on the formula's structure. However, we encounter difficulties in the case of $f = \mu Z.g$. Because, even if we can show that $\vdash g \rightarrow \hat{g}$ by induction, the formula $\mu Z.\hat{g}$ might not be disjunctive.
- To circumvent this issue, Walukiewicz[2] introduced the notion "*tableau consequence*", which is weaker than tableau equivalence. The key observation is that when g_1 is an aconjunctive formula and g_2 is a disjunctive one, $g_1 \rightarrow g_2$ is provable if some tableau of g_2 is a tableau consequence of that of g_1.

However, adapting this technique to show the completeness of PμTL looks infeasible. As one may have noticed, for our tableaux, we do not have a concrete distribution from a [MOD]-node to its children. This feature serves to avoid infinitary branching in tableau construction — with our approach, we are assured to have finitely many possible choices to continue the construction from [MOD]-nodes. However, this leads to difficulties in defining tableau equivalence and/or tableau consequence for PμTL. The reason is, in both notions, only literals can be exhibited. Thus, we cannot acquire sufficient information for the probabilistic extension.

Therefore, for PμTL, to present a complete axiom system for the whole logic might require the development of new tools instead of tableau consequence. We leave this challenging task as our future work.

Acknowledgment

W. W. Liu is supported by the National Science Foundation of China (No. 61872371), the Open Fund from the State Key Laboratory of High Performance Computing of China (HPCL) (No. 2020001-07), and the National Key Research and Development Program of China (No. 2018YFB0204301). L. J. Zhang is supported by the Guangdong Science and Technology (No. 2018B010107004) and the National Science Foundation of China (Nos. 61761136011 and 61836005).

References

[1] D. Kozen, Results on the propositional μ-calculus, *Theor. Comput. Sci.*, vol. 27, no. 3, pp. 333–354, 1983.

[2] I. Walukiewicz, Completeness of Kozen's axiomatisation of the propositional μ-calculus, *Inf. Comput.*, vol. 157, nos. 1–2, pp. 142–182, 2000.

[3] B. Banieqbal and H. Barringer, Temporal logic with fixed points, in *Temporal Logic in Specification*, B. Banieqbal, H. Barringer, and A. Pnueli, eds. Berlin, Germany: Springer, 1987, pp. 62–74.

[4] C. Morgan and A. McIver, A probabilistic temporal calculus based on expectations, in *Proc. Formal Methods Pacific*, Singapore, 1997, pp. 4–22.

[5] D. Niwiński and I. Walukiewicz, Games for the μ-calculus, *Theor. Comput. Sci.*, vol. 163, nos. 1–2, pp. 99–116, 1996.

[6] K. Tamura, Completeness of Kozen's axiomatization for the modal mu-calculus: A simple proof, arXiv preprint arXiv: 1408.3560, 2014.

[7] X. Song, W. Jiang, X. Liu, H. Lu, Z. Tian, and X. Du, A survey of game theory as applied to social networks, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 734–742, 2020.

[8] D. Kozen, Semantics of probabilistic programs, *J. Comput. Syst. Sci.*, vol. 22, no. 3, pp. 328–350, 1981.

[9] H. Hansson and B. Jonsson, A logic for reasoning about time and reliability, *Formal Asp. Comput.*, vol. 6, no. 5, pp. 512–535, 1994.

[10] C. Baier, On algorithmic verification methods for probabilistic systems, Habilitationsschrift, Fakultät für Mathematik und Informatik, Universität Mannheim, Mannheim, Germany, 1998.

[11] C. Baier and J. P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: The MIT Press, 2008.

[12] A. Bianco and L. de Alfaro, Model checking of probabilistic and nondeterministic systems, in *Foundations of Software Technology and Theoretical Computer Science*, P. S. Thiagarajan, ed. Berlin, Germany: Springer, 1995, pp. 499–513.

[13] T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera, The satisfiability problem for probabilistic CTL, in *Proc. 2008 23rd Annu. IEEE Symp. Logic in Computer Science*, Pittsburgh, PA, USA, 2018, pp. 391–402.

[14] L. de Alfaro, Formal verification of probabilistic systems, PhD dissertation, Stanford University, Stanford, CA, USA, 1997.

[15] R. Dimitrova, L. M. F. Fioriti, H. Hermanns, and R. Majumdar, Probabilistic CTL*: The deductive way, in *Tools and Algorithms for the Construction and Analysis of Systems*, M. Chechik and J. F. Raskin, eds. Berlin, Germany: Springer, 2016, pp. 280–296.

[16] W. W. Liu, L. Song, J. Wang, and L. J. Zhang, A simple probabilistic extension of modal μ-calculus, in *Proc. 24th Int. Conf. Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 882–888.

[17] Z. B. He and J. X. Zhou, Inference attacks on genomic data based on probabilistic graphical models, *Big Data Mining and Anal ytics*, vol. 3, no. 3, pp. 225–233, 2020.

[18] L. de Alfaro and R. Majumdar, Quantitative solution of omega-regular games, in *Proc. 33rd Annu. ACM Symp. Theory of Computing*, New York, NY, USA, 2001, pp. 675–683.

[19] M. Huth and M. Kwiatkowska, Quantitative analysis and model checking, in *Proc. 20th Annu. IEEE Symp. Logic in Computer Science*, Warsaw, Poland, 1997, pp. 111–122.

[20] A. K. McIver and C. C. Morgan, Games, probability and the quantitative μ-calculus $qM\mu$, in *Logic for Programming, Artificial Intelligence, and Reasoning*, M. Baaz and A. Voronkov, eds. Berlin, Germany: Springer, 2002, pp. 292–310.

[21] A. McIver and C. Morgan, Results on the quantitative μ-calculus $qM\mu$, *ACM Trans. Comput. Log.*, vol. 8, no. 1, p. 3, 2007.

[22] M. Mio, Probabilistic modal μ-calculus with independent product, in *Foundations of Software Science and Computational Structures*, M. Hofmann, ed. Berlin, Germany: Springer, 2011, pp. 290–304.

[23] R. Cleaveland, S. P. Iyer, and M. Narasimha, Probabilistic temporal logics via the modal mu-calculus, *Theor. Comput. Sci.*, vol. 342, nos. 2&3, pp. 316–350, 2005.

[24] P. F. Castro, C. Kilmurray, and N. Piterman, Tractable probabilistic μ-calculus that expresses probabilistic temporal logics, in *Proc. 32nd Int. Symp. Theoretical Aspects of Computer Science: STACS'15*, E. W. Mayr and N. Ollinger, eds. Dagstuhl, Germany, 2015, pp. 211–223.

[25] S. Chakraborty and J. P. Katoen. On the satisfiability

of some simple probabilistic logics, in *Proc. 31st Annu. ACM/IEEE Symp. Logic in Computer Science*, New York, NY, USA, 2016, pp. 56–65.

[26] F. Song, Y. D. Zhang, T. L. Chen, Y. Tang, and Z. W. Xu, Probabilistic alternating-time μ-calculus, in *Proc. 33rd AAAI Conf. Artificial Intelligence, AAAI 2019, The 31st Innovative Applications of Artificial Intelligence Conf., IAAI 2019, The 9th AAAI Symp. Educational Advances in Artificial Intelligence, EAAI 2019*, Honolulu, HI, USA, 2019, pp. 6179–6186.

[27] K. G. Larsen, R. Mardare, and B. T. Xue, Probabilistic mu-calculus: decidability and complete axiomatization, in *Proc. 36th IARCS Annu. Conf. Foundations of Software Technology and Theoretical Computer Science: FSTTCS*, A. Lal, S. Akshay, S. Saurabh, and S. Sen, eds. Dagstuhl, Germany, 2016, pp. 25:1–25:18.

[28] R. S. Streett and E. A. Emerson, An automata theoretic decision procedure for the propositional mu-calculus, *Inf. Comput.*, vol. 81, no. 3, pp. 249–264, 1989.

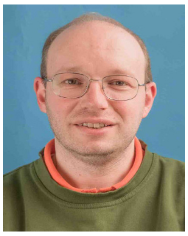
[29] J. Xu, W. Liu, D. N. Jansen, and L. Zhang, An axiomatisation of the probabilistic μ-calculus, in *Formal Methods and Software Engineering*, Y. A. Ameur and S. C. Qin, eds. Cham, Germany: Springer, 2019, pp. 420–437.

[30] C. Löding, Methods for the transformation of ω-automata: complexity and connection to second order logic, Master dissertation, Christian-Albrechts-University of Kiel, Kiel, Germany, 1998.

Wanwei Liu received the bachelor and PhD degrees from National University of Defense Technology (NUDT) in 2002 and 2009, respectively. He is now an associate professor at National University of Defense Technology. His research interests include automata theory, temporal logic, game theory, model checking, and probabilistic systems. He also works on other topics on formal methods and software engineering.

Junnan Xu received the bachelor degree from Peking University in 2016, and received the MSc degree from the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Science in 2020. His main research interests include temporal logic, axiomatisation, and reasoning.

David N. Jansen received the master in mathematics from the University of Bern, Switzerland in 1997, and the PhD degree in computer science from the University of Twente, The Netherlands in 2003. After his postdoctoral research in Saarbrücken and Aachen, Germany, he became an assistant professor at the Radboud University Nijmegen, The Netherlands. Since 2016, he has been at the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. He is interested in research on formal methods for probabilistic systems.



Andrea Turrini received the MSc and PhD degrees in computer science from the University of Verona, Italy in 2005 and 2009, respectively. He was first a postdoctoral researcher in Verona and then at Saarland University, Saarbrücken, Germany. He then moved to the Institute of Software, Chinese Academy of Sciences, Beijing, China as an associate research professor in the group of Lijun Zhang. His research interests include the formal analysis of probabilistic and quantum systems, with a particular focus on system minimization and verification of omega regular properties. He is also interested in omega regular automata, including learning-based algorithms for automata operations.

Lijun Zhang is a research professor at State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences (ISCAS). He is also the director of the Sino-Europe Joint Institute of Dependable and Smart Software at the Institute of Intelligent Software in Guangzhou. Before this he was a postdoctoral researcher at University of Oxford, and then an associate professor at Language-Based Technology section, DTU Compute, Technical University of Denmark. He gained the diploma and PhD (Dr. Ing.) degrees from Saarland University in 2003 and 2008, respectively. His research interests include probabilistic model checking, simulation reduction and decision algorithms, abstraction and model checking, learning algorithms, and verification of deep neural networks. He is leading the development of several tools including PRODeep for verifying neural networks, ROLL for learning automata, and the model checker ePMC, previously known as IscasMC.