A Simple Probabilistic Extension of Modal Mu-calculus

Wanwei Liu[1], Lei Song[2], Ji Wang[1], and Lijun Zhang[3]

[1] School of Computer Science,
National University of Defense Technology, Changsha, P. R. China
[2] Center of Quantum Computation and Intelligence Systems,
University of Technology, Sydney, Australia
[3] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

Abstract. Probabilistic systems are an important theme in AI domain. As the specification language, PCTL is the most frequently used logic for reasoning about probabilistic properties. In this paper, we present a natural and succinct probabilistic extension of μ-calculus, another prominent logic in the concurrency theory. We study the relationship with PCTL. Surprisingly, the expressiveness is highly orthogonal with PCTL. The proposed logic captures some useful properties which cannot be expressed in PCTL. We investigate the model checking and satisfiability problem, and show that the model checking problem is in **UP** ∩**co-UP**, and the satisfiability checking can be decided via reducing into solving parity games. This is in contrast to PCTL as well, whose satisfiability checking is still an open problem.

1 Introduction

Temporal logics are heavily used in theoretical computer science and AI-related fields. Among those, modal μ-calculus receives a lot of attraction ever since Kozen's seminal work [20]. See for example, [2,19,31,3]. Moreover, various temporal logics including LTL [26], CTL [12], CTL* [13] are extensively studied. It is known that their expressiveness is strictly less [10] than μ-calculus (aka. μTL), and their model checking algorithm has been proposed: for CTL the problem can be solved in polynomial time, whereas for LTL the problem is **PSPACE**-complete [29].

Probabilistic systems, such as Markov chains and Markov decision processes, are an important theme in AI domain. To reason about properties for probabilistic systems, the logic CTL was first extended with probabilistic quantifiers in [16] , resulting in the logic PCTL. Intuitively, $(a\mathsf{U}^{\geq 0.9}b)$ means that the probability of reaching b-states along a-states is at least 0.9. At the same time, probabilistic LTL and its extension PCTL* have all been studied. As in the classical setting, model checking problem for PCTL can be solved in polynomial time, whereas only exponential algorithms are known for LTL [9]. There have also been several attempts to extend μTL with probabilities in the literature. As we shall discuss in the related work, the extensions are either highly nontrivial in terms of the complexity of the corresponding model checking and satisfiability problems, or hindered from the restriction of fixpoint nesting.

We propose a natural and succinct extension of μTL in this paper, and name it PμTL. The logic is acquired by equipping the next operator with probability quantifiers, and keeping other parts as standard μTL. We have for instance the formula $\nu Z.(a \wedge \mathsf{X}^{\geq 0.8} Z)$. We investigate the model checking, expressiveness, and satisfiability problems of PμTL.

In detail, we first investigate the model checking problem of PμTL upon Markov chains. It turns out to be a straightforward adaptation of the classical algorithms for μTL, and the complexity remains in **UP** $\cap$ **co-UP**. We then give a comprehensive study on the expressiveness of PμTL by comparing with PCTL, and prove that PμTL is orthogonal with PCTL in expressiveness. However, for the qualitative fragments (i.e., probabilities may appear in a formula are only 0 and 1), we show that qualitative PμTL is strictly more expressive (w.r.t. finite Markov chains). On the other side, the satisfiability checking is quite challenging: we exploit the notion of probabilistic alternating parity automata (PAPA, for short), and reduce the Satisfiability problem into the Emptiness problem of PAPA. Further, this is reduced to solving parity games, and it is shown that both of these two problems are in 2**EXPTIME**. This is in contrast to PCTL as well, whose Satisfiability checking is still an open problem (cf. [6,4]).

An illustrating example We introduce a running example to motivate our work: Suppose there is a hacker trying to attack a remote server. The hacker has a supercomputer at hand and is trying to guess the password in a brute-force manner. For simplicity, we assume the password is a sequence of l letters, each of which is from '0'-'9', 'a'-'z', and 'A'-'Z'. Therefore, the total number of possible passwords is $n = 62^l$. The hacker let the supercomputer randomly generate a password, and see whether the decryption succeeds. If yes, the hacker wins; otherwise he tries with another one. However, if the supercomputer generates three wrong passwords in a row, it will be blocked for a certain amount of time until it can start another round of attacking — assuming that the password may be changed during the blocked moment, hence it does not make sense for the supercomputer to store all generated passwords. The whole process is illustrated in Fig. 1. Starting from s_1, we can see that the probability of eventually reaching *attacked*, i.e., the hacker decrypts successfully, equal 1, no matter how big l is (hence, the PCTL formula $\mathsf{F}^{\geq 1}$*attacked* holds), and we may conclude that the system is unsafe — this is of course against our intuition, as such system is considered to be safe if l is big enough. However, as we will show later, all PCTL formulae are not capable of expressing this property. By making use of PμTL, such property of security can be characterized easily as follows: $\nu Z.(\neg attacked \wedge \mathsf{X}^{\geq p} Z))$ with $p = {}^{n-3}/_{n-2}$, where $\neg$*attacked* denotes all other states in Fig. 1 different from s_5.

Motivation from AI perspective The presented logic has the following potential application in AI domain:

- First of all, Markov chains and Markov decision processes are the basic models in several areas of AI. As a logic with semantics defined w.r.t. such models, it could definitely be used in designating probability-relevant properties upon them. Particularly, the properties that could not be expressed by PCTL.
- Motion planing is an important topic in AI area, where standard μTL has once been adopted [5], because of its powerful expressiveness and the decidability of its Satisfiability problem. Thus, we expect that PμTL could be used in stochastic motion planning — since, PμTL is a decidability-preserving extension of μTL.

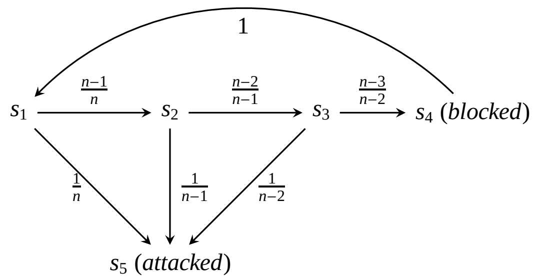

Fig. 1. An illustration of the hacking process

– Fixpoints play an important role in mathematics and computer science. In AI area, it is used to designate non-terminating behaviors of intelligent systems, such as maintenance goals [28]. Fixpoints act as the elementary ingredients in PμTL, hence such logic can also be used in such a situation.

Related work Probabilistic extensions of μTL have been studied by many authors: e.g., μ-calculi proposed in [25,17,11,21,22,24] interpret a formula as a function from states to real values in $[0, 1]$, whose semantics is different from PμTL. A further extension of μ-calculus was proposed in [23], which is able to encode the full PCTL. However, the model checking and Satisfiability algorithms are still unknown for these calculi and are "far from trivial" [24]. The other probabilistic μ-calculus was introduced in [8] along with a model checking algorithm for it. Moreover, it is able to encode PCTL formulae as well. However, that calculus only allows alternation-free formulae (cf. [14]).

Very recently — and independently —, Castro, Kilmurray, and Piterman present another extension by adding fixpoints to full PCTL [7]. The calculus they introduced is more expressive than logics PCTL and PCTL*. Moreover, it is also easy to see that it is a proper super logic of our logic PμTL as well. They show the model checking problem is in **NP** $\cap$**co-NP**. We note that some examples in our paper are similarly investigated in [7]. Since the logic in [7] subsumes PCTL, its Satisfiability problem is also left open. However in this paper we show Satisfiability of PμTL could be reduced to solving parity games, which makes this problem solvable in 2**EXPTIME**.

2 Preliminaries

In this paper, we fix a countable set $\mathcal{A}$ of *atomic propositions*, ranging over a, b, a_1 etc, and fix a countable set $\mathcal{Z}$ of *formula variables*, ranging over Z, Z_1 etc.

A *Markov chain* is a tuple $M = (S, \boldsymbol{T}, L)$, where S is a finite set of *states*; $\boldsymbol{T} : S \times S \to [0, 1]$ is the matrix of transition-probabilities, fulfilling $\sum_{s' \in S} \boldsymbol{T}(s, s') = 1$ for every $s \in S$; and $L : S \to 2^{\mathcal{A}}$ is the labeling function. A *pointed Markov chain* is a pair (M, s) where M is a Markov chain $(S, \boldsymbol{T}, L)$ and $s \in S$ is the *initial state*.

An (infinite) *path* π of M is an infinite sequence of states $s_0, s_1, \ldots$, such that $s_i \in S$ and $\boldsymbol{T}(s_i, s_{i+1}) > 0$ for each i. A basic *cylinder* $\mathrm{cyl}(s_0, s_1, \ldots, s_n)$ of M is the set of infinite paths having $s_0, s_1, \ldots, s_n$ as the prefix.

According to the standard theory of Markov process, the pointed Markov chain (M, s) uniquely derives a *measure space* $(\Pi_{M,s}, \Delta_{M,s}, \mathrm{prob}_{M,s})$ where $\Pi_{M,s}$ consists of all infinite paths of M; $\Delta_{M,s}$ is the minimal Borel field containing all basic cylinder of M (i.e., $\Delta_{(M,s)}$ is closed under complementation and countable intersection); and the measuring function $\mathrm{prob}_{M,s}$ fulfills: $\mathrm{prob}_{M,s}(\mathrm{cyl}(s_0, s_1, \ldots, s_n))$ equals 0 if $s \neq s_0$, and equals $\prod_{i<n} \boldsymbol{T}(s_i, s_{i+1})$ otherwise. We say a set $P \subseteq \Pi_{M,s}$ is *measurable* if $P \in \Delta_{M,s}$. [30] shows that the intersection of $\Pi_{M,s}$ and an omega-regular set must be measurable.

The syntax of PCTL formulae is described by the following abstract grammar:

$$f ::= \top \mid \bot \mid a \mid \neg a \mid \mathsf{X}^{\sim p} f \mid f \wedge f \mid f \vee f \mid f \mathsf{U}^{\sim p} f \mid f \mathsf{R}^{\sim p} f$$

where $\sim \in \{>, \geq\}$ and $p \in [0, 1]$. We also abbreviate $\top \mathsf{U}^{\sim p} f$ and $\bot \mathsf{R}^{\sim p} f$ as $\mathsf{F}^{\sim p} f$ and $\mathsf{G}^{\sim p} f$, respectively.

Semantics of a PCTL formula is given w.r.t. a Markov chain. For each PCTL formula f and a Markov chain $M = (S, \boldsymbol{T}, L)$, we will use $[\![f]\!]_M$ to denote the subset of S satisfying f, inductively defined as follows.

- $[\![\top]\!]_M = S$; $[\![\bot]\!]_M = \emptyset$.
- $[\![a]\!]_M = \{s \in S \mid a \in L(s)\}$; $[\![\neg a]\!]_M = \{s \in S \mid a \notin L(s)\}$.
- $[\![\mathsf{X}^{\sim p} f]\!]_M = \{s \in S \mid \sum_{s' \in [\![f]\!]_M} \boldsymbol{T}(s, s') \sim p\}$.
- $[\![f_1 \wedge f_2]\!]_M = [\![f_1]\!]_M \cap [\![f_2]\!]_M$; $[\![f_1 \vee f_2]\!]_M = [\![f_1]\!]_M \cup [\![f_2]\!]_M$.
- $[\![f_1 \mathsf{U}^{\sim p} f_2]\!]_M = \{s \in S \mid \mathrm{prob}_{M,s}\{\pi \in \mathrm{cyl}(s) \mid \pi \models f_1 \mathsf{U} f_2\} \sim p\}$ and $[\![f_1 \mathsf{R}^{\sim p} f_2]\!]_M = \{s \in S \mid \mathrm{prob}_{M,s}\{\pi \in \mathrm{cyl}(s) \mid \pi \models f_1 \mathsf{R} f_2\} \sim p\}$.

In addition, for an infinite path $\pi = s_0, s_1, \ldots$ of M, the notation $\pi \models f_1 \mathsf{U} f_2$ stands for that there is some $i \geq 0$ such that $s_i \in [\![f_2]\!]_M$ and $s_j \in [\![f_1]\!]_M$ for each $j < i$. Meanwhile, $\pi \models f_1 \mathsf{R} f_2$ holds if either $\pi \models f_2 \mathsf{U}(f_1 \wedge f_2)$ or $s_j \in [\![f_2]\!]_M$ for each j. To simplify notations, in what follows we denote by $M, s \models f$ whenever $s \in [\![f]\!]_M$ holds.

3 PμTL, Syntax and Semantics

In this section we present a simple probabilistic extension of modal μ-calculus, called PμTL. The syntax of PμTL formulae is depicted as follows:

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

Semantics of a PμTL formula is given w.r.t. a Markov chain $M = (S, \boldsymbol{T}, L)$ and an assignment $e : \mathcal{Z} \to 2^S$. Similarly, for each PμTL formula f, we denote by $[\![f]\!]_M(e)$ the state set satisfying f under e. Inductively:

- $[\![\top]\!]_M(e) = S$ and $[\![\bot]\!]_M(e) = \emptyset$.
- $[\![a]\!]_M(e) = \{s \in S \mid a \in L(s)\}$ and $[\![\neg a]\!]_M(e) = \{s \in S \mid a \notin L(s)\}$.
- $[\![Z]\!]_M(e) = e(Z)$.
- $[\![\mathsf{X}^{\sim p} f]\!]_M(e) = \{s \in S \mid \sum_{s' \in [\![f]\!]_M(e)} \boldsymbol{T}(s, s') \sim p\}$.
- $[\![f_1 \wedge f_2]\!]_M(e) = [\![f_1]\!]_M(e) \cap [\![f_2]\!]_M(e)$
  and $[\![f_1 \vee f_2]\!]_M(e) = [\![f_1]\!]_M(e) \cup [\![f_2]\!]_M(e)$.
- $[\![\mu Z.f]\!]_M(e) = \bigcap\{S' \subseteq S \mid [\![f]\!]_M(e[Z \mapsto S']) \subseteq S'\}$ and $[\![\nu Z.f]\!]_M(e) = \bigcup\{S' \subseteq S \mid [\![f]\!]_M(e[Z \mapsto S']) \supseteq S'\}$.

Indeed, $[\![\mu Z.f]\!]_M(e)$ (resp. $[\![\nu Z.f]\!]_M(e)$) could be computed as in the classical setting via the following iteration:

1. let $S_0 = \emptyset$ (resp. $S_0 = S$);
2. subsequently, let $S_{i+1} = [\![f]\!]_M(e[Z \mapsto S_i])$;
3. stops if $S_{\ell+1} = S_\ell$, and returns S_ℓ.

Note that the algorithm obtains a monotonic chain with such an iteration, and hence it must terminate within finite steps. Actually, $[\![\mu Z.f]\!]_M(e)$ (resp. $[\![\nu Z.f]\!]_M(e)$) captures the least (resp. greatest) solution of $X = [\![f]\!]_M(e[Z \mapsto X])$ within 2^S.

Semantical definition of PμTL formulae also yields the model checking algorithm.

Theorem 1. *The model checking problem of PμTL is in* ***UP*** $\cap$***co-UP****.*

Indeed, the proof is analogous to the non-probabilistic version [18,32] and the only noteworthy difference lies from handling $\mathsf{X}^{\sim p}$- subformulae, opposing to $\Box$- and $\Diamond$-subformulae, which could be proceeded in (deterministic) polynomial time.

In what follows, we directly denote by $[\![f]\!]_M$ in the case that f is a closed formula (i.e., each variable of f is bound), and we also denote by $M, s \models f$ if $s \in [\![f]\!]_M$.

Below we give some example properties:

(1) The formula $\nu Z.(a \wedge \mathsf{X}^{>0.8} Z)$ describes that there exists an a-region, where each state has less than 0.2 probability to escape from it immediately (i.e., in one step).
(2) $\nu Z.(a \wedge \mathsf{X}^{>0}\mathsf{X}^{>0} Z)$ says that there is a cycle in the Markov chain, such that a holds at least in every even step.
(3) $M, s \models \mu Z.(a \vee \mathsf{X}^{\geq 0.6} Z)$ if some a-state is reachable from s, but at each step, one just has some probability (not less than 0.6) to go on with the right direction.
(4) The PμTL formula $\mu Z.(b \vee (a \wedge \mathsf{X}^{\geq 1} Z))$ holds if $a\mathsf{U}b$ holds along each path. It is stronger than the property described by the PCTL formula $a\mathsf{U}^{\geq 1}b$. For the latter allows the existence of a-cycles.
(5) As a more complicated example, the formula $\nu Z_1.(a \vee \mu Z_2.(a \vee \mathsf{X}^{>0} Z_2) \wedge \mathsf{X}^{\geq 1} Z_1)$ just tells the story that "a will be surely encountered", as described by $\mathsf{F}^{\geq 1} a$ with PCTL.

Given a PμTL formula f and a bound variable Z, we use $\mathscr{D}_f(Z)$ to denote the sub-formula which binds Z in f. For example, let $f = \mu Z_1(a \wedge \nu Z_2.(b \wedge \mathsf{X}^{>=0.3} Z_2) \vee \mathsf{X}^{>0.6} Z_1)$, then we have $\mathscr{D}_f(Z_1) = f$ and $\mathscr{D}_f(Z_2) = \nu Z_2.(b \wedge \mathsf{X}^{\geq 0.3} Z_2)$.

We say that a PμTL formula f is *guarded*, if the occurrence of each bound variable Z in $\mathscr{D}_f(Z)$ is in the scope of some X-operator. The following theorem could be proven in a same manner as that in [31].

Theorem 2. *For each PμTL formula f, there is a guarded formula f' such that $[\![f']\!]_M(e) = [\![f]\!]_M(e)$ for every M and e.*

Thus, in what follows, we always assume that each PμTL formula is guarded.

4 Expressiveness

In this section, we will give a comparison between PμTL and PCTL, and we are only concerned about closed PμTL formulae. For a PμTL formula f and a PCTL formula g, we say that f and g are *equivalent* if $[\![f]\!]_M = [\![g]\!]_M$ for every Markov chain M, denoted as $f \equiv g$.

First of all, we will show that some PμTL formula could not be equivalently expressed by any PCTL formula.

Theorem 3. *Let $f = \nu Z.(a \wedge \mathsf{X}^{\geq 0.5} Z)$, then $g \not\equiv f$ for every PCTL formula g.*

Proof. To show this, we need first construct two families of Markov chains, namely, $M_0, M_1, \ldots$, and $M'_0, M'_1, M'_2, \ldots$.

For the first group, let $M_n = (\{s_0, s_1, \ldots, s_n\}, \boldsymbol{T}_n, L_n)$, where: $\boldsymbol{T}_n(s_0, s_0) = 1$ and $\boldsymbol{T}_n(s_{i+1}, s_i) = 1$ for each $i < n$ (hence $\boldsymbol{T}_n(s_i, s_j) = 0$ for any other s_i, s_j). In addition, $L_n(s_0) = \emptyset$ and $L_n(s_i) = \{a\}$ for each $0 < i \leq n$.

For the second ones, let $M'_n = (\{s'_0, s'_1, \ldots, s'_n\}, \boldsymbol{T}'_n, L'_n)$ where: $\boldsymbol{T}'_n(s'_n, s'_n) = \boldsymbol{T}'_n(s'_n, s'_{n-1}) = 0.5$, $\boldsymbol{T}'_n(s_0, s_0) = 1$, and $\boldsymbol{T}'_n(s'_{i+1}, s'_i) = 1$ for every $i < n - 1$. In addition, $L'_n(s'_0) = \emptyset$ and $L'_n(s'_i) = \{a\}$ for each $0 < i \leq n$.

Given a PCTL formula g, let $N(g)$ be the maximal nesting depth of temporal-operators of g. According to [1, Thm. 10.45], we have that $M'_n, s'_n \models g$ if and only if $M_n, s_n \models g$ whenever $n \geq N(g)$.

Observe the fact that $M'_n, s'_n \models f$ and $M_n, s_n \not\models f$ for every $n \geq 1$. Assume that there exists some PCTL formula g fulfilling $f \equiv g$, then we have

$$\begin{aligned} M'_{N(g)}, s'_{N(g)} \models f &\iff M'_{N(g)}, s'_{N(g)} \models g \\ \iff M_{N(g)}, s_{N(g)} \models g &\iff M_{N(g)}, s_{N(g)} \models f \end{aligned}$$

and hence it results in a contradiction. ◻

Conversely, the following theorem reveals that there also exists some PCTL formula that could not be equivalently expressed by any PμTL formula.

Theorem 4. *Let $f = \mathsf{F}^{\geq 0.5} a$, then $g \not\equiv f$ for every (closed) PμTL formula g.*

Proof. Let $M = (\{s_1, s_2, s_3\}, \boldsymbol{T}, L)$ be the (family of) Markov chain(s) where: $L(s_1) = L(s_2) = \emptyset$, $L(s_3) = \{a\}$, $\boldsymbol{T}(s_1, s_1) = x, \boldsymbol{T}(s_1, s_2) = y, \boldsymbol{T}(s_1, s_3) = z$, and $\boldsymbol{T}(s_2, s_2) = \boldsymbol{T}(s_3, s_3) = 1$, with $x, y, z \in (0, 1)$ and $x + y + z = 1$.

For every PCTL and/or closed PμTL formula g, we let $P_x(g)$ be the proposition that "for the fixed x, there are infinitely many y making $M, s_1 \models g$ and there are infinitely many y making $M, s_1 \not\models g$". We now show that if g is a closed PμTL formula, then there exists some $x_g < 1$ such that $P_x(g)$ does not hold whenever $x \in (x_g, 1)$.

- Such x_g can be arbitrarily chosen if $g = \bot$, $g = \top$, $g = a$ or $g = \neg a$.
- In the case that $g = g_1 \wedge g_2$, assume by contradiction that such x_g does not exist, then it implies that for every $x \in (0, 1)$, there exists some $x' > x$ such that $P_{x'}(g)$ holds. Observe that $M, s_1 \models g$ implies both $M, s_1 \models g_1$ and $M, s_1 \models g_2$; and $M, s_1 \not\models g$ implies either $M, s_1 \not\models g_1$ or $M, s_1 \not\models g_2$. Thus, we can infer that either x_{g_1} or x_{g_2} does not exist, which violates the induction hypothesis.

– Proof for the case of $g = g_1 \vee g_2$ is similar to the above.
– If $g = \mathsf{X}^{\sim p} g'$ and $p \in (0,1)$, whenever $x \in (\max\{p, 1-p\}, 1)$, since $\sim \in \{>, \geq\}$, then $M, s_1 \models g$ iff $M, s_1 \models g'$ because $y + z < p$ in such situation. In this case, we may just let $x_g = \max\{x_{g'}, p, 1-p\}$.
– If $g = \mathsf{X}^{\geq 1} g'$, then we need to distinguish two cases: 1) There exist $x, y \in (0,1)$ such that $M, s_1 \models g$ holds, then we can immediately infer that both $M, s_2 \models g'$ and $M, s_3 \models g'$. In addition, observe that truth values of g' on s_2 and s_3 are irrelevant to x and y. It implies that in such case $M, s_1 \models g$ iff $M, s_1 \models g'$, and hence, we may just let $x_g = x_{g'}$. 2) There is no such x and y having $M, s_1 \models g$ holds, in such situation, x_g can be any number in $(0,1)$.
– If $g = \mathsf{X}^{>0} g'$, then the proof is similar to the above.
– When $g = \mathsf{X}^{\geq 0} g'$ (or $g = \mathsf{X}^{>1} g'$), things would be trivial, because g could be reduced to $\top$ (resp. $\bot$) in such case.
– If $g = \mu Z.g'$, we let $g_0 = \bot$ and $g_{i+1} = g'[Z/g_i]$. Since that M is a 3-state Markov chain, then g and $\bigvee_{i \leq 3} g_i$ share the same truth value at every state of M. This indicates that all least fix-points could be eliminated w.r.t. such Markov chain.
– When $g = \nu Z.g'$, the preprocessing is almost similar, but we just replace g with $\bigwedge_{i \leq 3} g_i$ where $g_0 = \top$.

Now, for the PCTL formula $f = \mathsf{F}^{\geq 0.5} a$, such x_f does not exist, because, for every $x \in (0,1)$ we have: $M, s_1 \models f$ provided that $y \in [(1-x)/2, 1)$; and $M, s_1 \not\models f$ if $y \in (0, (1-x)/2)$. This implies that $P_x(f)$ holds for every $x \in (0,1)$, and hence f cannot be equally expressed by any PμTL formula. □

Note that the value 0.5 in the previous two theorems can be generalized to any other probability $p \in (0,1)$.

We also provide a comparison on the *qualitative* fragments of PCTL and PμTL. Probabilities occurring in such fragments can only be 0 or 1.

Theorem 5. *Every qualitative PCTL formula can be equally expressed by a qualitative PμTL formula.*

Proof. We will give a constructive translation procedure, which takes a qualitative PCTL formula g and outputs an equivalent qualitative PμTL formula $\widetilde{g}$. Inductively:

1. $\widetilde{g} = \bot$ if $g = \bot$, or its root operator is $\mathsf{X}^{>1}$, $\mathsf{U}^{>1}$ or $\mathsf{R}^{>1}$;
   $\widetilde{g} = \top$ if $g = \top$, or its root operator is $\mathsf{X}^{\geq 0}$, $\mathsf{U}^{\geq 0}$ or $\mathsf{R}^{\geq 0}$.
2. $\widetilde{g} = \widetilde{g_1} \wedge \widetilde{g_2}$ if $g = g_1 \wedge g_2$; and $\widetilde{g} = \widetilde{g_1} \vee \widetilde{g_2}$ if $g = g_1 \vee g_2$.
3. $\widetilde{g} = \mathsf{X}^{>0} \widetilde{g'}$ if $g = \mathsf{X}^{>0} g'$; and $\widetilde{g} = \mathsf{X}^{\geq 1} \widetilde{g'}$ if $g = \mathsf{X}^{\geq 1} g'$.
4. $\widetilde{g} = \mu Z.(\widetilde{g_2} \vee (\widetilde{g_1} \wedge \mathsf{X}^{>0} Z))$ if $g = g_1 \mathsf{U}^{>0} g_2$;
   and $\widetilde{g} = \nu Z.(\widetilde{g_2} \wedge (\widetilde{g_1} \vee \mathsf{X}^{\geq 1} Z))$ if $g = g_1 \mathsf{R}^{\geq 1} g_2$.
5. $\widetilde{g} = \nu Z.(\widetilde{g_2} \vee (\widetilde{g_1} \wedge \widetilde{\mathsf{F}^{>0} g_2} \wedge \mathsf{X}^{\geq 1} Z)) = \nu Z.(\widetilde{g_2} \vee (\widetilde{g_1} \wedge \mu Z'.(\widetilde{g_2} \vee \mathsf{X}^{>0} Z') \wedge \mathsf{X}^{\geq 1} Z))$ if $g = g_1 \mathsf{U}^{\geq 1} g_2$;
   and $\widetilde{g} = \mu Z.(\widetilde{g_2} \wedge (\widetilde{g_1} \vee \widetilde{\mathsf{G}^{\geq 1} g_2} \vee \mathsf{X}^{>0} Z)) = \mu Z.(\widetilde{g_2} \wedge (\widetilde{g_1} \vee \nu Z'.(\widetilde{g_2} \wedge \mathsf{X}^{\geq 1} Z') \vee \mathsf{X}^{>0} Z))$ if $g = g_1 \mathsf{R}^{>0} g_2$.

The proof of equivalence could be done by induction on the structure of the formula. □

Note that Thm. 5 holds because we are only concerned about finite models in this paper. Interested readers may show that it is not true for infinite Markov chains.

Theorem 6. *The qualitative PμTL formula* $f = \nu Z.(a \wedge \mathsf{X}^{>0}\mathsf{X}^{>0}Z)$ *cannot be expressed in qualitative PCTL.*

Proof. Construct a series of Markov chains $M''_2, M''_3, \ldots$ such that each M''_n is the Markov chain $(\{s''_0, s''_1, \ldots, s''_n\}, \boldsymbol{T}''_n, L''_n)$, where $\boldsymbol{T}''_n(s''_0, s''_0) = 1$ and $\boldsymbol{T}''_n(s''_{i+1}, s''_i) = 1$ for each $i < n$. In addition, $L''_n(s''_i) = \{a\}$ for each $i \neq 1$, and $L''_n(s''_1) = \emptyset$.

For a given PCTL formula g, let $\hat{g}$ be the LTL formula obtained from g by discarding all probability quantifiers, e.g., we have $\hat{g} = a\mathsf{U}(b \vee \mathsf{G}\neg a)$ if $g = a\mathsf{U}^{\geq 0.3}(b \vee \mathsf{G}^{>0.6}\neg a)$. Since that from s''_n the Markov chain M''_n has exactly one infinite path $\pi_n = s''_n, \ldots, s''_1, (s''_0)^\omega$, then for each $n \geq 2$ we have $M''_n, s''_n \models g$ if and only if $\pi_n \models \hat{g}$. It is shown in [33] that $M''_n, s''_n \models \hat{g}$ iff $M''_{n+1}, s''_{n+1} \models \hat{g}$ in the case of $n \geq N'(\hat{g}) = N'(g)$, where $N'(g)$ and $N'(\hat{g})$ are the nesting depth of X-operator of g and $\hat{g}$, respectively. Thus, we have $M''_n, s''_n \models g$ iff $M''_{n+1}, s''_{n+1} \models g$ in such situation. This implies that $\nu Z.(a \wedge \mathsf{X}^{>0}\mathsf{X}^{>0}Z)$ has no equivalent qualitative PCTL expression, because we cannot simultaneously have $M''_n, s''_n \models f$ and $M''_{n+1}, s''_{n+1} \models f$ for each $n \geq 2$. □

Note that the conclusion of Thm. 6 is also pointed out in [8], and we here provide a detailed proof. Indeed, this proof also works for general PCTL formulae, and hence the property $\nu Z.(a \wedge \mathsf{X}^{>0}\mathsf{X}^{>0}Z)$ even cannot be expressed by any PCTL formula.

5 Automata Characterization

In this section, we will define a new type of automata recognizing (pointed) Markov chains, called *probabilistic alternating parity automata* (PAPA, for short), and such automata could be viewed as the probabilistic extension of those defined in [32].

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 *initial state*, δ is the *transition function* to be defined later, and $\Omega : Q \rightsquigarrow \mathbb{N}$, is a partial function of *coloring*; in what follows, we say a state is *colored* if Ω is defined for the state.

The notion of *transition conditions* over Q is inductively defined as follows:

1. $\perp$ and $\top$ are transition conditions over Q.
2. For every $a \in \mathcal{A}$, the literals a and $\neg a$ are transition conditions over Q.
3. If $q \in Q$, then q is a transition condition over Q.
4. If $q \in Q$ and $p \in [0, 1]$, then $\bigcirc^{\sim p} q$ is a transition condition over Q, where $\sim \in \{\geq, >\}$.
5. If $q_1, q_2 \in Q$ then both $q_1 \vee q_2$ and $q_1 \wedge q_2$ are transition conditions over Q.

The transition function δ assigns each state $q \in Q$ a transition condition over Q.

We denote by R_A the *derived graph* of A, its vertex set is just Q, and there is an edge from q_1 to q_2 iff q_2 appears in $\delta(q_1)$. We say that A is *well-structured*, if for every path $q_1, q_2, \ldots, q_n$ that forms a cycle (i.e., $q_1 = q_n$) in R_A, we have that: 1) there exists some $1 \leq i < n$ such that $\delta(q_i) = \bigcirc^{\sim p} q_{i+1}$ with some $p \in [0, 1]$; 2) there exists some $1 \leq j < n$ such that q_j is colored. In what follows, we are only concerned about well-structured PAPA.

Given a pointed Markov chain (M, s_0) with $M = (S, \boldsymbol{T}, L)$ and $s_0 \in S$, a *run* of A over (M, s_0) is a $Q \times S$-labeled tree (T, λ) fulfilling: $\lambda(v_0) = (q_0, s_0)$ for the root vertex v_0; and for each internal vertex v of T with $\lambda(v) = (q, s)$ we require that

- $\delta(q) \neq \bot$, and if $\delta(q) = \top$ then v has no child;
- $a \in L(s)$ if $\delta(q) = a$, and $a \notin L(s)$ if $\delta(q) = \neg a$;
- if $\delta(q) = q_1 \wedge q_2$ then v has two children v_1 and v_2 respectively having $\lambda(v_1) = (q_1, s)$ and $\lambda(v_2) = (q_2, s)$;
- if $\delta(q) = q_1 \vee q_2$ then v has one child v' with $\lambda(v') \in \{(q_1, s), (q_2, s)\}$;
- v has one child v' having $\lambda(v') = (q', s)$, if $\delta(q) = q'$;
- if $\delta(q) = \bigcirc^{\sim p} q'$ then v has a set of children $v_1, \ldots, v_n$ such that $\lambda(v_i) = (q', s_i)$, where $\sum_{i=1}^{n} \boldsymbol{T}(s, s_i) \sim p$.

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

$$\max\{\, n \mid \text{there are infinitely many } i \text{ s.t. } \Omega(\mathrm{proj}_1(\lambda(v_i))) = n\}$$

where $\mathrm{proj}_1(q, s) = q$. A run (T, λ) is *accepting* if n_τ is an even number, for every infinite branch τ of T. A pointed Markov chain (M, s_0) is *accepted* by A if A has an accepting run over it. We denote by $\mathscr{L}(A)$ the set consisting of pointed Markov chains accepted by A.

Theorem 7. *Given a closed PμTL formula f, there is a PAPA A_f such that: $M, s \models f$ iff $(M, s) \in \mathscr{L}(A_f)$, for each pointed Markov chain (M, s).*

Proof. We just let $A_f = (Q_f, q_f, \delta_f, \Omega_f)$, where:

- $Q_f = \{q_g \mid g \text{ is a subformula of } f\}$, and hence $q_f \in Q_f$;
- 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 $\delta_f(q_{g_1 \vee g_2}) = q_{g_1} \vee q_{g_2}$;
  - $\delta_f(q_{\mathsf{X}^{\sim p} g}) = \bigcirc^{\sim p} q_g$;
  - $\delta_f(q_{\mu Z.g}) = q_g$ and $\delta_f(q_{\nu Z.g}) = q_g$;
  - $\delta_f(q_Z) = q_{\mathscr{D}_f(Z)}$.
- Ω_f is defined at every state q_Z with $Z \in \mathcal{Z}$ fulfilling: If Z is a μ-variable (resp. ν-variable), then $\Omega_f(q_Z)$ is the minimal odd (resp. even) number which is greater than every $\Omega_f(q_{Z'})$ such that $\mathscr{D}_f(Z')$ is a subformula of $\mathscr{D}_f(Z)$.

It could be directly examined that A_f is well-structured since f is guarded. The proof of equivalence can be similarly done as that in [32] — the only different induction step is to deal with transitions being of $\bigcirc^{\sim p} q$ (in that paper, the corresponding cases are $\Box q$ and $\Diamond q$). Actually, we can see that if a PAPA (Q, q, δ, Ω) corresponds to the PμTL formula g, then the PAPA $(Q \cup \{q'\}, q', \delta[q' \mapsto \bigcirc^{\sim p} q], \Omega)$ must correspond to $\mathsf{X}^{\sim p} g$. $\square$

6 Satisfiability Decision

It is known from Section 5 that the SATISFIABILITY problem of PμTL could be reduced to the EMPTINESS problem of PAPA. In this section, we will further reduce it to parity game solving.

A *parity game* G is a tuple (V, E, C), where: V is a finite set of *locations*, and V could be partitioned into two disjoint sets V^0 and V^1; $E \subseteq V \times V$ is the set of *moves*, required to be total; and $C : V \rightsquigarrow \mathbb{N}$ is a partial function of *coloring*, and we say a location v is *colored*, if $C(v)$ is defined. In addition, for the game G, we require that each loop involves at least one colored location.

Two players — player 0 and player 1, are respectively in charge of V^0 and V^1 when G is being played. A *play* of G starting from $v_0 \in V$ is an infinite sequence of locations $v_0, v_1, \ldots$ made by player 0 and player 1 — for every $i \in \mathbb{N}$, the location v_{i+1} is chosen by player 0 (resp. player 1) with $(v_i, v_{i+1}) \in E$ whenever $v_i \in V^0$ (resp. $v_i \in V^1$).

Player 0 (resp. player 1) *wins* the play $v_0, v_1, \ldots$ if the maximal color occurring infinitely often in it is even (resp. odd) — and we say that a color c *occurs* in this play if there is some v_i with $C(v_i) = c$.

A *winning strategy* for player i is a mapping $H_i : V^* \cdot V^i \to V$, such that for every play $v_0, v_1, \ldots$, player i always wins if $v_{j+1} = H_i(v_0, \ldots, v_j)$ whenever $v_j \in V^i$. In addition, H_i is *memoryless* if $H_i(v_0, \ldots, v_j)$ agrees with $H_i(v_j)$ for every j.

Theorem 8 ([15,34,18]). *For a parity game G, from every location, there is exactly one player having a winning strategy. The problem of deciding the winner at a location is in **UP** $\cap$**co-UP**. In addition, if a player has a winning strategy then she also has a memoryless one from the same location.*

We use $\mathscr{W}_i(G)$ to denote the set consisting of all locations from which player i has a winning strategy.

Given a PAPA $A = (Q, q, \delta, \Omega)$, a *gadget* D of A is a finite directed acyclic digram (P, γ) where $P \subseteq Q$, $\gamma \subseteq P \times P$, and for each $q \in P$:

1. if $\delta(q) = q'$, then $q' \in P$ and $(q, q') \in \gamma$;
2. if $\delta(q) = q_1 \wedge q_2$ then $q_1, q_2 \in P$, and $(q, q_1), (q, q_2) \in \gamma$;
3. if $\delta(q) = q_1 \vee q_2$ then there is some $i \in \{1, 2\}$ such that $q_i \in P$ and $(q, q_i) \in \gamma$,
4. q has no successor for the other cases.

For convenience, we sometimes directly write $q \in D$ whenever $D = (P, \gamma)$ and $q \in P$. We denote by $\mathcal{D}(A)$ the set consisting of all gadgets of A. Since we require that each PAPA A is well-structured, then $\mathcal{D}(A)$ must be a finite set.

Given a sequence of gadgets $D_1, D_2, \ldots$ such that $D_i = (P_i, \gamma_i)$, an *infinite path* within it is a sequence of states $q_{1,1}, \ldots, q_{1,\ell_1}, q_{2,1}, \ldots, q_{2,\ell_2}, \ldots$ such that each $(q_{i,j}, q_{i,j+1}) \in \gamma_i$ and $\delta(q_{i,\ell_i}) = \bigcirc^{\sim p_i} q_{i+1,1}$ for some $p_i \in [0, 1]$. We say such an infinite path is *even* (resp. *odd*) if the maximal color (w.r.t. Ω) occurring infinitely often is even (resp. odd).

We say that a gadget $D = (P, \gamma)$ is *incompatible* if there exist $q_1, q_2 \in P$ and $\delta(q_1) = a$, $\delta(q_2) = \neg a$ for some $a \in \mathcal{A}$; or there is some $q \in P$ with $\delta(q) = \perp$. Otherwise, we say that D is *compatible*.

Let D be a gadget and $\Gamma = \{D_1, \ldots, D_k\}$ be a set of gadgets, we denote by $\Gamma \Vdash D$ if there exist k positive numbers $x_1, \ldots, x_k$ such that: $\sum_{i=1}^{k} x_i \leq 1$, and for each $q \in D$

with $\delta(q) = \bigcirc^{\sim p} q'$, we have $\sum_{q' \in D_i} x_i \sim p$. We in what follows call $x_1, \ldots, x_k$ the *enabling condition*. Note that the relation $\Vdash$ could be decided by solving a linear system of inequality.

According to automata theory, we may construct a deterministic (word) parity automaton $\widetilde{A} = (\widetilde{Q}, \widetilde{q}, \widetilde{\delta}, \widetilde{\Omega})$ were $\widetilde{\delta} : \widetilde{Q} \times \mathcal{D}(A) \to \widetilde{Q}$ and $\widetilde{\Omega}$ is a total coloring function. It takes a gadget sequence as input, and accepts it if every gadget in it is compatible and every infinite path within it is even.

Then, we may create a parity game $G_A = (V_A, E_A, C_A)$ for the PAPA A, in detail:

– $V_A = V_A^0 \cup V_A^1$, where $V_A^0 = 2^{\mathcal{D}(A) \times \widetilde{Q}}$ and $V_A^1 = \mathcal{D}(A) \times \widetilde{Q}$.
– $E_A = \{(\{(D_1, \widetilde{q_1}), \ldots, (D_k, \widetilde{q_k})\}, (D_i, \widetilde{q_i})) \mid 1 \le i \le k\} \cup$
  $\{((D, \widetilde{q}), \{(D_1, \widetilde{q_1}), \ldots, (D_k, \widetilde{q_k})\}) \mid (D_1, \ldots, D_k) \Vdash D,$
  and each $\widetilde{q_i} = \widetilde{\delta}(\widetilde{q}, D_i)\}$.
– $C_A(D, \widetilde{q}) = \widetilde{\Omega}(\widetilde{q})$, hence every location in V_A^1 is colored.

Theorem 9. *Let the PAPA $A = (Q, \underset{\sim}{q}, \delta, \Omega)$, then $\mathcal{L}(A) \neq \emptyset$ if and only if there is some $D \in \mathcal{D}(A)$ with $q \in D$ such that $\{(D, \widetilde{\delta}(\widetilde{q}, D))\} \in \mathcal{W}_0(G_A)$.*

Proof. $\Longrightarrow$) Suppose that there is some pointed Markov chain $(M = (S, \boldsymbol{T}, L), s) \in \mathcal{L}(A)$, then there exists some accepting run (T, λ) of A on (M, s).

We say a vertex v of T is a *modal vertex* if $\delta(\mathrm{proj}_1(\lambda(v)))$ is of the form $\bigcirc^{\sim p} q'$. We denote by $\|v\|$ the *modal depth* of v, i.e., the number of modal vertices among the ancestors of v.

From each vertex v of T, we may obtain a set of vertices, denoted as $\mathrm{cls}(v)$, which involves v and all its descendants with the same modal depth. Since A is well-structured, then $\mathrm{cls}(v)$ must be a finite set. We also lift the notation by defining $\mathrm{cls}\, V = \bigcup_{v \in V} \mathrm{cls}(v)$ for a finite vertex set V.

In addition, each finite vertex set V of T derives a gadget $\mathbf{D}(V) = (P_V, \gamma_V)$, where $P_V = \{\mathrm{proj}_1(\lambda(v)) \mid v \in V\}$, and $(q_1, q_2) \in \gamma_V$ if there are two vertices $v_1, v_2 \in V$, such that $\mathrm{proj}_1(\lambda(v_i)) = q_i$ for $i = 1, 2$ and v_2 is a child of v_1.

Let v_0 be the root vertex of T, then we have $\lambda(v_0) = (q, s)$. We now let $D = D_0 = \mathbf{D}(\mathrm{cls}(v_0))$, then for each play $\Delta_0, (D_0, \widetilde{q_0}), \Delta_1, (D_1, \widetilde{q_1}), \Delta_2, \ldots$ with $\Delta_0 = (D, \widetilde{\delta}(\widetilde{q}, D))$ and each $D_i = (P_i, \gamma_i)$, player 0 can control it and make the play to fulfill the following property:

(*) For each i, there exists a finite set of vertices V_i having the same modal depth i, and there exists a state s_i of M; and $q' \in P_i$ iff there is some $v_{q'} \in V_i$ such that $\lambda(v_{q'}) = (q', s_i)$. In addition, $(q_1, q_2) \in \gamma_i$ iff v_{q_2} is a child of v_{q_1}.

For $i = 0$, we have $V_0 = \mathrm{cls}(v_0)$ and $s_0 = s$. Assume that (*) holds at step i, then player 0 chooses the next location guided by the run as following: First, let V_i' be all modal vertices among V_i, and let V_i'' be the set consisting of children of vertices in V_i'. Then, V_i'' can be partitioned into several sets $V_{i,1}'', \ldots, V_{i,k}''$ according to the second component (assume $\mathrm{proj}_2(\lambda(v')) = s_{i,j}$ for $v' \in V_{i,j}''$) labeled on the vertices. Player 0 then chooses the set $\{(D_{i,1}, \widetilde{q_{i,1}}), \ldots, (D_{i,k}, \widetilde{q_{i,k}})\}$ as the next location, where $D_{i,j} = \mathbf{D}(\mathrm{cls}(V_{i,j}''))$ and $\widetilde{q_{i,j}} = \widetilde{\delta}(\widetilde{q_i}, D_{i,j})$.

Then, according to the construction, for each $(D_{i,j}, \widetilde{\delta}(\widetilde{q_{i,j}}))$ we have some state $s_{i,j}$ and the vertex set $\mathrm{cls}(V''_{i,j})$ making property (*) holds, no matter how player 1 chooses. Let $x_1 = \boldsymbol{T}(s_i, s_{i,1}), \ldots, x_k = \boldsymbol{T}(s_i, s_{i,k})$, we definitely have $\sum_{j=1}^{k} x_j \leq 1$ and we also have $\sum_{q'' \in D_{i,j}} x_j \sim p$ for each $q' \in D_i$ such that $\delta(q') = \bigcirc^{\sim p} q''$ because (T, λ) is an accepting run. Therefore, $(D_{i,1}, \ldots, D_{i,k}) \Vdash D_i$ holds.

We assert that each $D_i = (q_1, \ldots, q_\ell)$ must be compatible — since (T, λ) is accepting, no such $q' \in D_i$ having $\delta(q') = \bot$, and if there exist $q_1, q_2 \in D_i$ with $\delta(q_1) = a$ and $\delta(q_2) = \neg a$, then we will both have $a \in L(s_i)$ and $a \notin L(s_i)$. Also note that each infinite path within $D_0, D_1, \ldots$ corresponds to the first component of the labelings of an infinite branch of T, hence it must be even. According to $\widetilde{A}$, we then conclude that this strategy is winning for player 0 form $\{(D_0, \widetilde{\delta}(\widetilde{q}))\}$.

$\Longleftarrow$) Let H_0 be the (memoryless) winning strategy of player 0 from $\{(D, \widetilde{\delta}(\widetilde{q}))\}$, where D is some gadget involving q. We say that a location $l = (D^l, \widetilde{q^l}) \in V^1_{\widetilde{A}}$ is *feasible* if l may appear in some play under control of player 0 according to H_0. We create a Markov chain $M = (S, \boldsymbol{T}, L)$ as follows.

- First, let $S = \{s_l \mid l \text{ is a feasible location}\} \cup \{s'\}$.
- Second, since each feasible location must be compatiable, then we may let $L(s_l) = \{a \in \mathcal{A} \mid \text{there is some } q' \text{ in } D^l\}$. Meanwhile, we let $L(s') = \emptyset$.
- The transition matrix $\boldsymbol{T}$ is determined as follows: For each feasible location l, suppose that $H_0(l) = \{l_1 = (D^{l_1}, \widetilde{q^{l_1}}), \ldots, l_k = (D_k, \widetilde{q^{l_k}})\}$, since $\{D^{l_1}, \ldots, D^{l_k}\} \Vdash D^l$ then we have a set of enabling condition $x_1, \ldots, x_k$. We let $\boldsymbol{T}(s_l, s_{l_j}) = x_j$ for each j, let $\boldsymbol{T}(s_l, s') = 1 - \sum_{j=1}^{k} x_j$, and let $\boldsymbol{T}(s', s') = 1$.

What left is to show that $(M, s_{l_0}) \in \mathscr{L}(A)$, where l_0 is just $(D, \widetilde{\delta}(\widetilde{q}))$. For each gadget D^l such that l is feasible, we could obtain a forest (T_l, λ_l), and in which each vertex q' is labeled with (q', s_l). Then from T_{l_0} (which is an exact tree with (q, s_{l_0}) labeled in the root), with a top-down manner, we connect the so far added tree T_l with every $T_{l'}$ such that $l' \in H_0(l)$ — i.e., for each q' in T_l with $\delta(q') = \bigcirc^{\sim p} q''$, we add the vertex q'' in $T_{l'}$ as a child — it can be seen that it must be the case that some edges connecting some leaves of T_l and the root(s) of $T_{l'}$. We denote the labeled tree finally get as (T, λ), and it is indeed be an accepting run of A over (M, s_{l_0}). □

Intuitively, player 0 could extract a winning strategy from an accepting run of A over any pointed Markov chain; and conversely, one can construct a pointed Markov chain accepted by A according to the (memoryless) winning strategy of player 0.

As a consequence of Thm. 7, Thm. 8 and Thm. 9 we have the following main conclusion of this section.

Theorem 10. *Both the* EMPTINESS *problem of PAPA and the* SATISFIABILITY *problem of PμTL are decidable, and both of them are in 2**EXPTIME**.*

Indeed, from Thm. 7 one can get a PAPA whose scale is linear in the size of the input formula, and an n-state PAPA could be converted to a parity game with scale $2^{2^{O(n)}}$. From standard game theory (see [18,32], and see [27] for an improved bound), and with a similar analysis of [32] (see also the analysis of the coloring number in that paper), one can infer that this problem is in 2**EXPTIME**.

7 Discussion

In this paper, we present the logic PμTL, a simple and succinct probabilistic extension of μTL. We have compared the expressiveness of these two kinds of logics: In general, PμTL captures 'local' and 'stepwise' probabilities; whereas PCTL could describe 'global' probabilities in the system. Hence, these two logics are orthogonal and complementary, and one can obtain a more powerful and expressive logic by combing them together, as done in [7]. i.e., we may use formulae like $(\mu Z.(a \vee \mathsf{X}^{\geq 0.8} Z))\mathsf{U}^{\geq 0.6}(\nu Z'.(b \wedge \mathsf{F}^{>0.3} Z'))$. Model checking algorithm of such an extension can be acquired from those of the underlying logics.

In this paper, we have also investigated the decision problem of PμTL, the key issue and the most challenging part is to deal with probabilistic quantifiers when doing reduction to parity games, which is a highly nontrivial extension of the non-probabilistic case. As a cost, we have only now got an algorithm with double-exponential time complexity for solving it — in contrast, the SATISFIABILITY problem for the standard μTL is in **EXPTIME**.

Acknowledgement

First and foremost, the authors would thank all the anonymous reviewers for the valuable and helpful comments on this paper. We would also thank Nir Piterman for his valuable comments on our work.

Wanwei Liu is supported by National Natural Science Foundation of China (Grant Nos. 61103012, 61379054 and 61272335). Lei Song is supported by Australian Research Council under Grant DP130102764. Ji Wang is supported by National Natural Science Foundation of China (Grant No. 61120106006). Lijun Zhang (corresponding author) is supported by National Natural Science Foundation of China (Grant Nos. 61428208, 61472473 and 61361136002), the CAS/SAFEA International Partnership Program for Creative Research Teams.

References

1. C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
2. B. Banieqbal and H. Barringer. Temporal logic with fixed points. In *B. Banieqbal, H. Barringer, and A. Pnueli, editors, Temporal Logic in Specification*, volume 398 of *Lecture Notes in Computer Science*, pages 62–74. Springer-Verlag, 1987.
3. S. Berezin. *Model Checking and Theorem Proving: A Unified Framework*. Phd thesis, Carnegie Mellon University, Pittsburgh, PA, USA, Jan. 2002.
4. N. Bertrand, J. Fearnley, and S. Schewe. Bounded satisfiability for PCTL. In *CSL*, volume 16 of *LIPIcs*, pages 92–106. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
5. A. Bhatia, M. R. Maly, L. E. Kavraki, and M. Y. Vardi. Motion planning with complex goals. *Robotics & Automation Magazine, IEEE*, 18(3):55–64, 2011.
6. T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera.
7. P. Castro, C. Kilmurray, and N. Piterman. Tractable probabilistic μ-calculus that expresses probabilistic temporal logics. In *32nd Symposium on Theoretical Aspects of Computer Science*, volume 30 of *Leibniz International Proceedings in Informatics*, pages 211–223, 2015.

8. R. Cleaveland, S. Iyer, and M. Narasimha. Probabilistic temporal logics via the modal μ-calculus. *Theor. Comput. Sci.*, 342(2-3):316–350, 2005.
9. J.-M. Couvreur, N. Saheb, and G. Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *LPAR 2003*, volume 2850 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2003.
10. M. Dam. Translating CTL* into the modal μ-calculus. Technical Report ECS-LFCS-90-123, Laboratory for Foundations of Computer Science, University of Edinburgh, November 1995.
11. L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. In *STOC*, pages 675–683. ACM, 2001.
12. E. Emerson and E. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Proc. of the 7th Int. Colloquium on Automata, Languages and Programming (ICALP'80)*, volume 85 of *Lecture Notes in Computer Science*, pages 169–181. Springer-Verlag, 1980.
13. E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logics of branching time. *Journal of the ACM*, 33(1):151–178, 1986.
14. E. A. Emerson and C. L. Lei. Efficient model checking in fragments of the propositional mu-calculus. In *First IEEE Symposium on Logic in Computer Science,* pages 267–278. Los Alamitos: IEEE Computer Society, 1986.
15. Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proceeding of 14th ACM Symposium on the Theory of Computing*, pages 60–65, San Francisco, California, 1982.
16. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *FAC*, 6(5):512–535, 1994.
17. M. Huth and M. Z. Kwiatkowska. Quantitative analysis and model checking. In *LICS*, pages 111–122. IEEE Computer Society, 1997.
18. M. Jurdziński. Deciding the winner in parity games is in **UP**∩co-**UP**. *Information Processing Letters*, 68(3):119–124, 1998.
19. J.-P. Katoen. *Concepts, Algorithms, and Tools for Model Checking*. FAU, Lehrstuhl für Informatik VII Friedrich-Alexander Universität Erlangen-Nürnberg, 2 edition, 1998. Lecture Notes of the Course "Mechanised Validation of Parallel Systems".
20. D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
21. A. McIver and C. Morgan. Games, probability and the quantitative μ-calculus qmμ. In *LPAR*, volume 2514 of *Lecture Notes in Computer Science*, pages 292–310. Springer, 2002.
22. A. McIver and C. Morgan. Results on the quantitative μ-calculus qmμ. *TOCL*, 8(1):3, 2007.
23. M. Mio. *Game semantics for probabilistic modal μ-calculi*. PhD thesis, The University of Edinburgh, 2012.
24. M. Mio. Probabilistic modal μ-calculus with independent product. *Logical Methods in Computer Science*, 8(4), 2012.
25. C. Morgan and A. McIver. A probabilistic temporal calculus based on expectations. In L. Groves and S. Reeves, editors, *Proc. Formal Methods Pacific*, pages 4–22. Springer, 1997.
26. A. Pnueli. The temporal logic of programs. In *Proc. of 18th IEEE Symposium on Foundation of Computer Science (FOCS' 77)*, pages 46–57. IEEE Computer Society, 1977.
27. S. Schewe. *Synthesis of Distributed Systems*. Phd thesis, Saarbrücken, 2008.
28. M. P. Singh. Applying the mu-calculus in planning and reasoning about action. *J. Log. Comput.*, 8(3):425–445, 1998.
29. A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of Assoc. Comput. Mach.*, 32(3):733–749, 1985.
30. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338. IEEE Computer Society, 1985.
31. I. Walukiewicz. Completeness of Kozen's axiomatization of the propositional μ-calculus. *Information and Computation*, 157:142–182, 2000.

32. T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull, Belg, Math, Soc*, 8(2):359–391, 2002.
33. P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.
34. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998.