Computation Tree Logic Model Checking Based on Possibility Measures

Yongming Li*, Yali Li, Zhanyou Ma

College of Computer Science, Shaanxi Normal University, Xi'an, 710062, China

Abstract

In order to deal with the systematic verification with uncertain infromation in possibility theory, Li and Li [19] introduced model checking of linear-time properties in which the uncertainty is modeled by possibility measures. Xue, Lei and Li [26] defined computation tree logic (CTL) based on possibility measures, which is called possibilistic CTL (PoCTL). This paper is a continuation of the above work. First, we study the expressiveness of PoCTL. Unlike probabilistic CTL, it is shown that PoCTL (in particular, qualitative PoCTL) is more powerful than CTL with respect to their expressiveness. The equivalent expressions of basic CTL formulae using qualitative PoCTL formulae are presented in detail. Some PoCTL formulae that can not be expressed by any CTL formulae are presented. In particular, some qualitative properties of repeated reachability and persistence are expressed using PoCTL formulae. Next, adapting CTL model-checking algorithm, a method to solve the PoCTL model-checking problem and its time complexity are discussed in detail. Finally, an example is given to illustrate the PoCTL model-checking method.

Keywords: Computation tree logic; possibilistic Kripke structure; possibility measure; qualitative property; quantitative property.

1. Introduction

Model checking [12] is a formal verification technique which allows for desired behavioral properties of a given system to be verified on the basis of a suitable model of the system through systematic inspection of all states of the model. It is widely used in the design and analysis of computer systems [6, 8]. Although

---

it has been rapidly gaining in importance in recent years, classical model checking can not deal with verification of those systems (e.g.,concurrent systems) dealing with uncertainty information. Such as, the development of most large and complex systems is inevitably involved with lots of uncertainty and inconsistency information.

In order to handle the systematic verification with uncertain information in probability, Hart and Sharir [15] in 1986 applied probability theory to model checking in which the uncertainty is modeled by probability measures. Baier and Katoen [1] systematically introduced the principle and method of model checking based on probability measures and related applications with Markov chain models for probabilistic systems. For the past few years, there were even more applications on probability model checking in verifying properties of systems with uncertain information (see e.g. [3]).

On the other hand, Zadeh proposed the theory of fuzzy sets in 1965 [28], and possibility measures [23, 29] are a development of classical measures as a branch of the theory of fuzzy sets from then. As a comparison, possibility measures (more general, fuzzy measures) focus on non-additive situation, while probability measures are used for additive situation. Most problems in real situations are complicated and non-additive. As a matter of fact, fuzziness seems to pervade most human perception and thinking processes as noted by Zadeh, especially, modeling human-centered systems, including biomedical systems ([20]), criminal trial systems, decision making systems([13]), linguistic quantifiers ([7, 27]), and knowledge base ([10]). Therefore, it is necessary to study the theory and its applications of model checking on non-deterministic systems of non-additive measure, especially, fuzzy measure. In this respect, Li and Li [19] introduced model checking of linear-time properties in which the uncertainty is modeled by possibility measures and initiated the model checking based on possibility measures. Xue, Lei and Li [26] defined computation tree logic based on possibility measures, which is called possibilistic computation tree logic (PoCTL, in short).

Although we have studied the quantitative and qualitative properties of PoCTL

in [26], there are many important issues that still have not been addressed. The first important problem is the expressiveness of PoCTL: whether any CTL formulae can be expressed by PoCTL or vise versa. As we know, probabilistic CTL and CTL are not comparable with each other ([1]). This allows probabilistic CTL to be used to do model checking of real-world problems, which can not be tackled by classical CTL model checking. The surprising result of this paper is that CTL is a proper subclass of PoCTL. The second problem is looking for the method to solve PoCTL model-checking problems. As we know, there are effective algorithms and automated tools to solve CTL model-checking problems. As we just mentioned, CTL is a proper subclass of PoCTL, it is nontrivial to study whether there are effective algorithms to solve the PoCTL model-checking problems. We shall give complete study to the above two problems in this paper.

The content of this paper is arranged as follows. In Section 2 we recall the notion of possibilistic Kripke structures, the related possibility measures induced by the possibilistic Kripke structures, and the main notions of PoCTL introduced in [26]. In Section 3, the equivalence of PoCTL formulae and CTL formulae is investigated, and the differences between PoCTL formulae and CTL formulae are discussed. An important result, CTL is a proper subclass of PoCTL, is obtained. Section 3 also presents qualitative properties of repeated reachability and persistence. The PoCTL model checking approach is presented in Section 4, and an illustrative example is given in Section 5. The paper ends with conclusion section.

2. Preliminaries

Transition systems or Kripke structures are key models for model checking. Corresponding to possibilistic model checking, we have the notion of possibilistic Kripke structures, which is defined as follows.

Definition 2.1. [19] A possibilistic Kripke structure is a tuple $M = (S, P, I, AP, L)$, where

(1) S is a countable, nonempty set of states;

(2) $P : S \times S \longrightarrow [0,1]$ is the transition possibility distribution such that for all states s, $\bigvee_{s' \in S} P(s, s') = 1$;

(3) $I : S \longrightarrow [0,1]$ is the initial distribution, such that $\bigvee_{s \in S} I(s) = 1$;

(4) AP is a set of atomic propositions;

(5) $L : S \longrightarrow 2^{AP}$ is a labeling function that labels a state s with those atomic propositions in AP that are supposed to hold in s.

Furthermore, if the set S and AP are finite sets, then $M = (S, P, I, AP, L)$ is called a finite possibilistic Kripke structure.

Remark 1. (1) In Definition 2.1, we require the transition possibility distribution and initial distribution are normal, i.e., $\vee_{s' \in S} P(s, s') = 1$ and $\vee_{s \in S} I(s) = 1$, where we use $\vee X$ or $\wedge X$ to represent the least upper bound (or supremum) or the largest lower bound (or infimum) of the subset $X \subseteq [0,1]$, respectively. These conditions are corresponding to the transition probability distribution and probability initial distribution in probabilistic Kripke structure or Markov chain ([1]), where the supremum operation is replaced by the sum operation. They are the main differences between possibilistic Kripke structure and probabilistic Kripke structure. In fact, in fuzzy uncertainty, the order instead of the additivity is one of the most important factors to be considered.

(2) The transition possibility distribution $P : S \times S \longrightarrow [0,1]$ can also be represented by a fuzzy matrix. For convenience, this fuzzy matrix is also written as P, i.e.,

$$P = (P(s,t))_{s,t \in S},$$

and P is also called the (fuzzy) transition matrix of M. In [19], we also used the symbol A to represent transition matrix. For the fuzzy matrix P, its transitive closure is denoted by P^+. When S is finite, and if S has N elements, i.e., $N = |S|$, then $P^+ = P \vee P^2 \vee \cdots \vee P^N$ [18], where $P^{k+1} = P^k \circ P$ for any positive integer number k. Here, we use the symbol $\circ$ to represent the max-min composition operation of fuzzy matrixes. Recall that the max-min composition operation of

4

fuzzy matrices is similar to ordinary matrix multiplication operation, that is, let ordinary multiplication and addition operations of real numbers be replaced by minimum and maximum operations of real numbers ([29]).

For a possibilistic Kripke structure $M = (S, P, I, AP, L)$, using P^+, we can get another possibilistic Kripke structure $M^+ = (S, P^+, I, AP, L)$.

(3) The authors in [14] also used the notion of fuzzy possibilistic Kripke structures as the models of qualitative possibilistic logic QFL, which is formally defined as a structure $K = (W, \Vdash, \pi)$ where W is a nonempty set of worlds, $\Vdash$ maps $AP \times W$ into the truth value set $\{0, 1/n, 2/n, \cdots, 1\}(n \geq 1)$, and π is a normalized positive fuzzy subset of W, i.e., a mapping $\pi : W \longrightarrow [0, 1]$ such that $\pi(w) > 0$ for each w and $\bigvee_{w \in W} \pi(w) = 1$. Obviously, the notion of fuzzy possibilistic Kripke structure just defined is not equivalent to our notion of possibilistic Kripke structures. Since our notion of possibilistic Kripke structures is obvious a generalization of classical Kripke structures (see [12]) into fuzzy cases and a possibilistic version of (discrete-time) Markov chains as defined in Definition 10.1 in [1]. So we still use the name of possibilistic Kripke structures here, but it has no connection with that defined in [14]. The much more related notion is (discrete-time) fuzzy Markov chains [17] or (discrete-time) possibilistic Markov chains ([10]) or possibilistic Markov processes ([16]) which are used to model certain fuzzy systems. The only difference between possibilistic Kripke structures and fuzzy (or possibilistic) Markov chains lies in that there is no labeling function in the definition of fuzzy (or possibilistic) Markov chains. In [10], possibilistic Markov chains are used to model the evolution of updating problem in a knowledge base that describes the state of evolving system. Uncertainty comes from incomplete knowledge about the knowledge base, "one may only have some idea about what is/are the most plausible state(s) of the system, among possible one"([10]). This type of incomplete knowledge was described in terms of possibility distribution in [10], the degree of transition possibility distribution denotes the plausible degree of the next state. This provides us a sort of justification for degrees of transitions in possibilistic Kripke structures.

5

The states s with $I(s) > 0$ are considered as the initial states. For state s and $T \subseteq S$, let $P(s, T)$ denote the possibility of moving from s to some state $t \in T$ in a single step, that is,

$$P(s, T) = \vee_{t \in T} P(s, t).$$

Paths in possibilistic Kripke structure M are infinite paths in the underlying digraph. They are defined as infinite state sequences $\pi = s_0 s_1 s_2 \cdots \in S^w$ such that $P(s_i, s_{i+1}) > 0$ for all $i \in I$. Let $Paths(M)$ denote the set of all paths in M, and $Paths_{fin}(M)$ denote the set of finite path fragments $s_0 s_1 \cdots s_n$ where $n \geq 0$ and $P(s_i, s_{i+1}) > 0$ for $0 \leq i \leq n$. Let $Paths_M(s)$ ($Paths(s)$ if M is understood) denote the set of all paths in M that start in state s. Similarly, $Paths_{M-fin}(s)$ ($Paths_{fin}(s)$ if M is understood) denotes the set of finite path fragments $s_0 s_1 \cdots s_n$ such that $s_0 = s$. The set of direct successors (called $Post$) and direct predecessors (named Pre) are defined as follows:

$$Post(s) = \{s' \in S \mid P(s, s') > 0\}; \quad Pre(s) = \{s' \in S \mid P(s', s) > 0\}.$$

Given a possibilistic Kripke structure M, the cylinder set of $\hat{\pi} = s_0 \cdots s_n \in Paths_{fin}(M)$ is defined as ([1])

$$Cyl(\hat{\pi}) = \{\pi \in Paths(M) | \hat{\pi} \in Pref(\pi)\},$$

where $Pref(\pi) = \{\pi' | \pi'$ is a finite prefix of $\pi\}$. Then as shown in [19], $\Omega = 2^{Paths(M)}$ is the algebra generated by $\{Cyl(\hat{\pi}) \mid \hat{\pi} \in Paths_{fin}(M)\}$ on $Paths(M)$. That is to say, $\Omega = 2^{Paths(M)}$ is the unique subalgebra of $2^{Paths(M)}$ which is closed under unions and intersections containing $\{Cyl(\hat{\pi}) | \hat{\pi} \in Pref(\pi)\}$.

Definition 2.2. [19] For a possibilistic Kripke structure M, a function $Po^M : Paths(M) \to [0, 1]$ is defined as follows:

$$Po^M(\pi) = I(s_0) \wedge \bigwedge_{i=0}^{\infty} P(s_i, s_{i+1}) \tag{1}$$

for any $\pi = s_0 s_1 \cdots , \pi \in Paths(M)$. Furthermore, we define

$$Po^M(E) = \vee\{Po^M(\pi) \mid \pi \in E\} \qquad (2)$$

for any $E \subseteq Paths(M)$, then, we have a well-defined function

$$Po^M : 2^{Paths(M)} \longrightarrow [0, 1],$$

Po^M is called the possibility measure over $\Omega = 2^{Paths(M)}$ as it has the properties stated in Theorem 2.1. If M is clear from the context, then M is omitted and we simply write Po for Po^M.

Theorem 2.1. *[19] Po is a possibility measure on $\Omega = 2^{Paths(M)}$, i.e., Po satisfies the following conditions:*

(1) $Po(\varnothing) = 0$, $Po(Paths(M)) = 1$;

(2) $Po(\bigcup\limits_{i \in I} A_i) = \bigvee\limits_{i \in I} Po(A_i)$ for any $A_i \in \Omega$, $i \in I$.

Theorem 2.2. *[19] Let M be a possibilistic finite Kripke structure. Then the possibility measure of the cylinder sets is given by $Po(Cyl(s_0 \cdots s_n)) = I(s_0) \wedge \bigwedge\limits_{i=0}^{n-1} P(s_i, s_{i+1})$ when $n > 0$ and $Po(Cyl(s_0)) = I(s_0)$.*

Remark 2. (1) For paths starting in a certain (possibly noninitial) state s, the same construction is applied to the possibilistic Kripke structure M_s that results from M by letting s be the unique initial state. Formally, for $M = (S, P, I, AP, L)$ and state s, M_s is defined by $M_s = (S, P, s, AP, L)$, where s denotes an initial distribution with only one initial state s.

(2) For a probabilistic Kripke structure M, by the intension property of probability measures, the induced probability measure ([1]), which is defined on the σ-algebra of $2^{Paths(M)}$ generated by cylinder sets, is uniquely determined by its definition on cylinder sets. On the other hand, by the extensional property of possibility measures, the induced possibility measure in Eq. (2) is uniquely determined by its definition on single paths as shown in Eq.(1). The method to define probability measure on a probabilistic Kripke structure can not be applied

7

to that of possibility measure on possibilistic Kripke structure, and vice versa. For more comparisons of possibility measures and probability measures, we refer to [9, 11, 13, 19] and references therein.

Definition 2.3. [26] (Syntax of PoCTL) *PoCTL state formulae* over the set AP of atomic propositions are formed according to the following grammar:

$$\Phi ::= \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid Po_J(\varphi)$$

where $a \in AP$, φ is a PoCTL path formula and $J \subseteq [0,1]$ is an interval with rational bounds.

   PoCTL path formulae are formed according to the following grammar:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \sqcup \Phi_2 \mid \Phi_1 \sqcup^{\leq n} \Phi_2$$

where Φ, Φ_1, and Φ_2 are state formulae and $n \in \mathbb{N}$.

Definition 2.4. [26] (Semantics of PoCTL) Let $a \in AP$ be an atomic proposition, $M = (S, P, I, AP, L)$ be a possibilistic Kripke structure, state $s \in S$, Φ, Ψ be PoCTL state formulae, and φ be a PoCTL path formula. *The satisfaction relation* $\models$ is defined *for state formulae* by

$$s \models a \qquad\qquad\qquad \text{iff } a \in L(s);$$
$$s \models \neg\Phi \qquad\qquad\qquad \text{iff } s \not\models \Phi;$$
$$s \models \Phi \wedge \Psi \qquad\qquad\qquad \text{iff } s \models \Phi \text{ and } s \models \Psi;$$
$$s \models Po_J(\varphi) \quad \text{iff } Po(s \models \varphi) \in J, \text{ where } Po(s \models \varphi) = Po^{M_s}(\{\pi \mid \pi \in Paths(s), \pi \models \varphi\}).$$

   For path π, *the satisfaction relation* $\models$ *for path formulae* is defined by

$$\pi \models \bigcirc\Phi \qquad\qquad\qquad \text{iff } \pi[1] \models \Phi;$$
$$\pi \models \Phi \sqcup \Psi \quad \text{iff } \exists k \geq 0, \pi[k] \models \Psi \text{ and } \pi[i] \models \Phi \text{ for all } 0 \leq i \leq k-1;$$
$$\pi \models \Phi \sqcup^{\leq n} \Psi \quad \text{iff } \exists 0 \leq k \leq n, (\pi[k] \models \Psi \wedge (\forall 0 \leq i < k), \pi[i] \models \Phi)).$$

where if $\pi = s_0 s_1 s_2 \cdots$, then $\pi[k] = s_k$ for any $k \geq 0$.

8

In particular, the path formulae $\Diamond\Phi$ ("eventually") and $\Box\Phi$ ("always") have the semantics

$$\pi = s_0 s_1 \cdots \models \Diamond\Phi \text{ iff } s_j \models \Phi \text{ for some } j \geq 0,$$

$$\pi = s_0 s_1 \cdots \models \Box\Phi \text{ iff } s_j \models \Phi \text{ for all } j \geq 0.$$

Alternatively, $\Diamond\Phi = \textit{true} \sqcup \Phi$.

Definition 2.5. [26] (Syntax of qualitative PoCTL) *State formulae in the qualitative fragment of PoCTL* (over AP) are formed according to the following grammar:

$$\Phi ::= \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid Po_{>0}(\varphi) \mid Po_{=1}(\varphi)$$

where $a \in AP$, φ is a path formula formed according to the following grammar:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \sqcup \Phi_2$$

where Φ, Φ_1 and Φ_2 are state formulae.

As a subclass of PoCTL, the semantics of qualitative PoCTL can be defined as that of PoCTL.

Since we shall compare the expressiveness of PoCTL and CTL, let us recall the definition of CTL.

Definition 2.6. [1](Syntax of CTL) State formulae in the fragment of CTL (over AP) are formed according to the following grammar:

$$\Phi ::= \textit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\varphi \mid \forall\varphi$$

where $a \in AP$, φ is a path formula formed according to the following grammar:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \sqcup \Phi_2$$

where Φ, Φ_1 and Φ_2 are state formulae.

9

Definition 2.7. [1] (Semantics of CTL) Let $a \in AP$ be an atomic proposition, $M = (S, P, I, AP, L)$ be a Kripke structure without terminal state (i.e., $\forall s \in S$, $\exists s' \in S$, $(s, s') \in P$), state $s \in S$, Φ, Ψ be CTL state formulae, and φ be a CTL path formula. The satisfaction relation $\models$ is defined for state formulae by

$$s \models a \qquad \text{iff } a \in L(s);$$
$$s \models \neg\Phi \qquad \text{iff } s \not\models \Phi$$
$$s \models \Phi \wedge \Psi \qquad \text{iff } s \models \Phi \text{ and } s \models \Psi;$$
$$s \models \exists\varphi \quad \text{iff } \pi \models \varphi \text{ for some } \pi \in Paths(s);$$
$$s \models \forall\varphi \quad \text{iff } \pi \models \varphi \text{ for all } \pi \in Paths(s).$$

For path π, the satisfaction relation $\models$ for path formulae is defined by

$$\pi \models \bigcirc\varphi \qquad \text{iff } \pi[1] \models \varphi;$$
$$\pi \models \Phi \sqcup \Psi \quad \text{iff } \exists k \geq 0, \pi[k] \models \Psi \text{ and } \pi[i] \models \Phi \text{ for all } 0 \leq i \leq k-1.$$

Remark 3. Since we use the PoCTL formula $Po_J(\varphi)$ to denote the possibility measure of the paths satisfying φ, i.e., $s \models Po_J(\varphi)$ iff $Po(s \models \varphi) \in J$, PoCTL is a possibility measure extension of classical CTL. Both the possibilistic and probabilistic CTL solve certain uncertainty of errors or other stochastic behaviors occurring in various real-world applications. As shown in [1], probabilistic CTL and CTL are not comparable with respect to their expressiveness. This allows probabilistic CTL to be used to solve the model-checking problems of real-world applications, which can not be tackled by classical model-checking algorithms. With regard to expressiveness of PoCTL, there was no further results on the comparisons between possibilistic CTL and classical CTL. We did not know whether PoCTL can express CTL or vise versa. We shall study the expressiveness of PoCTL in the next section and discuss PoCTL model checking then.

3. The expressiveness of PoCTL

In this section, we study how to define the equivalence between PoCTL formulae and CTL formulae. We intend to discuss the equivalence of PoCTL formulae

10

and CTL formulae and resolve the problem whether any PoCTL formula can be expressed by a CTL formula or not.

In this section, we always assume that M is a finite possibilistic Kripke structure.

Definition 3.1. For a possibilistic Kripke structure M with state space S, if Φ is a state formula, let $Sat_M(\Phi)$, or briefly $Sat(\Phi)$, denote $\{s \in S \mid s \models \Phi\}$.

Definition 3.2. PoCTL formulae Φ and Ψ are called equivalent, denoted $\Phi \equiv \Psi$, if $Sat(\Phi) = Sat(\Psi)$ for all finite possibilistic Kripke structures M over AP.

Definition 3.3. A PoCTL formula Φ is equivalent to a CTL formula Ψ, denoted $\Phi \equiv \Psi$, if $Sat_M(\Phi) = Sat_{TS(M)}(\Psi)$ for any finite possibilistic Kripke structure $M = (S, P, I, AP, L)$, where $TS(M) = (S, \rightarrow, I', AP, L)$ is defined by $s \rightarrow s'$ iff $Po(s, s') > 0$, and $s \in I'$ iff $I(s) > 0$. Obviously, $Paths_M(s) = Paths_{TS(M)}(s)$, wo we use the same symbol $Paths(s)$ to denote $Paths_M(s)$ and $Paths_{TS(M)}(s)$ in the following.

Remark 4. Definition 3.3 is a key notion, analogous to the one for probabilistic CTL. There are other ways to define an equivalence between CTL and PoCTL formulae. We shall give some discussion of this topic in Section 3.4.

Theorem 3.1. *Let $p \in [0, 1]$ be a rational number, φ an arbitrary PoCTL path formula, then, we have*

$$Po_{<p}(\varphi) \equiv \neg Po_{\geq p}(\varphi). \tag{3}$$

Proof. For any $p \in [0, 1]$, for any possibilistic Kripke structure M with state space S, we have

$$
\begin{aligned}
Sat(Po_{<p}(\varphi)) &= \{s \mid Po(s \models \varphi) < p\} \\
&= S - \{s \mid Po(s \models \varphi) \geq p\} \\
&= S - Sat(Po_{\geq p}(\varphi)) \\
&= Sat(\neg Po_{\geq p}(\varphi)).
\end{aligned}
$$

11

The last equality follows from the fact $Sat(\neg\Phi) = S - Sat(\Phi)$ for any PoCTL state formula Φ. Therefore, $Po_{<p}(\varphi) \equiv \neg Po_{\geq p}(\varphi)$. □

Dual to Theorem 3.1, we have

$$Po_{>p}(\varphi) \equiv \neg Po_{\leq p}(\varphi) \tag{4}$$

for any rational number $p \in [0, 1]$ and path formula φ. Then it is easy to prove that

$$Po_{(p,q)}(\varphi) \equiv \neg Po_{\leq p}(\varphi) \wedge \neg Po_{\geq q}(\varphi).$$

Although the qualitative fragment of PoCTL state formulae only allows possibility bounds of the form > 0 and $= 1$, bounds of the form $= 0$ and < 1 are also definable as

$$Po_{=0}(\varphi) \equiv \neg Po_{>0}(\varphi), \ Po_{<1}(\varphi) \equiv \neg Po_{=1}(\varphi).$$

3.1. CTL formulae are equivalent to PoCTL formulae

Theorem 3.2. *Let φ be any CTL path formula. Then, we have*

$$\exists\varphi \equiv Po_{>0}(\varphi). \tag{5}$$

Proof. Let M be a finite possibilistic Kripke structure, then we have $Sat_M(Po_{>0}(\varphi)) = \{s \mid Po(s \models \varphi) > 0\}$, and $Sat_{TS(M)}(\exists\varphi) = \{s \mid \exists\pi \in Paths(s), \pi \models \varphi\}$.

Assume $s \in Sat(Po_{>0}(\varphi))$, then, state s satisfies $Po(s \models \varphi) > 0$, and it follows that $\{s \mid \exists\pi \in Paths(s), \pi \models \varphi\} \neq \emptyset$, i.e., $s \in Sat_{TS(M)}(\exists\varphi)$. Therefore, $Sat_M(Po_{>0}(\varphi)) \subseteq Sat_{TS(M)}(\exists\varphi)$.

Conversely, if $s \in Sat_{TS(M)}(\exists\varphi)$, then $\exists\pi \in Paths(s), \pi \models \varphi$. Since M is finite and $\pi \in Paths(s)$, it follows that $Po^{M_s}(\pi) > 0$, and thus $Po(s \models \varphi) \geq Po^{M_s}(\pi) > 0$. Therefore, $s \in Sat_M(Po_{>0}(\varphi))$. This shows that $Sat_{TS(M)}(\exists\varphi) \subseteq Sat_M(Po_{>0}(\varphi))$.

The above shows that $Sat_{TS(M)}(\exists\varphi) = Sat_M(Po_{>0}(\varphi))$. Therefore, we have the required equality. □

To show the further relationship between CTL and PoCTL, we need the existential normal form of CTL formulae.

12

Definition 3.4. [1] For $a \in AP$, the set of CTL state formulae in existential normal form (ENF, in short) is given by

$$\Phi ::= true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \exists\bigcirc\Phi \mid \exists(\Phi_1 \sqcup \Phi_2) \mid \exists\square\Phi.$$

Theorem 3.3. *[1] For each CTL formulae there exists an equivalent CTL formulae in ENF.*

Theorem 3.4. *For any CTL formula, there exists an equivalent qualitative PoCTL formula.*

Proof. By Theorem 3.3, each CTL formula can be transformed into an equivalent formula in ENF. Then, by Theorem 3.2, each CTL formula in ENF is equivalent to a qualitative PoCTL formula. Combining Theorem 3.3 and Theorem 3.2, it follows that each CTL formula is equivalent to a qualitative PoCTL formula. $\square$

Theorem 3.4 shows that CTL is a subclass of PoCTL. We concretely write some equivalent formulae as follows, most of which do not hold in probabilistic CTL as declared in [1].

Proposition 3.1. *For any CTL formulae Φ and Ψ, we have*

(1) $\exists\Diamond\Phi \equiv Po_{>0}(\Diamond\Phi)$,

(2) $\exists\bigcirc\Phi \equiv Po_{>0}(\bigcirc\Phi)$,

(3) $\exists\square\Phi \equiv Po_{>0}(\square\Phi)$, and

(4) $\exists(\Phi \sqcup \Psi) \equiv Po_{>0}(\Phi \sqcup \Psi)$.

Proposition 3.2. *For any CTL formulae Φ and Ψ, we have*

(1) $\forall\bigcirc\Phi \equiv Po_{=0}(\bigcirc\neg\Phi)$,

(2) $\forall(\Phi \sqcup \Psi) \equiv Po_{=0}(\neg\Psi \sqcup (\neg\Phi \wedge \neg\Psi)) \wedge Po_{=0}(\square\neg\Psi)$,

(3) $\forall\Diamond\Phi \equiv Po_{=0}(\square\neg\Phi)$, and

(4) $\forall\square\Phi \equiv Po_{=0}(\Diamond\neg\Phi)$.

Remark 5. The above propositions may not hold in infinite possibilistic Kripke structure. We give a counterexample for Proposition 3.2 (3).

Assume Proposition 3.2 (3) holds in any infinite possibilistic Kripke structure M for $\Phi = a \in AP$, that is $\forall \Diamond a \equiv Po_{=0}(\Box \neg a)$ such that state s fulfills both the formula $Po_{=0}(\Box \neg a)$ and $\forall \Diamond a$ or none of them. Fig.1 gives an infinite possibilistic Kripke structure $M = (S, P, I, AP, L)$, in which states are represented by nodes and transitions by labeled edges. State names are depicted inside the ovals. Initial states are indicated by having an incoming arrow without source. We can see that $Paths(s_0) = \{s_0 s_1 s_2 \cdots s_k t^w | k \geq 0\}$. For this M, we have $Po(s_0 \models \Box \neg a) = \vee Po\{\pi \in Paths(s_0) \mid \pi \models \Box \neg a\} = 0$, and it follows that $s_0 \in Sat_M(Po_{=0}(\Box \neg a))$. But $s_0 s_1 s_2 \cdots \not\models \Diamond a$, i.e., $s_0 \notin Sat_{TS(M)}(\forall \Diamond a)$. This contradicts the assumption that $\forall \Diamond a \equiv Po_{=0}(\Box \neg a)$.
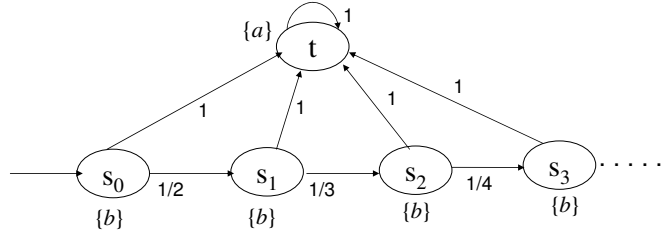

Fig.1.An infinite possibilistic Kripke structure M.

3.2. CTL is a proper subclass of PoCTL

Theorem 3.5. *There is no CTL formula that is equivalent to $Po_{=1}(\Diamond a)$.*

Proof. Assume that there is a CTL formula Φ such that $\Phi \equiv Po_{=1}(\Diamond a)$. Consider the following two finite possibilistic Kripke structures M_1 and M_2, see Fig.2 and Fig.3. By a simple calculation, we have $Po(s_0 \models \Diamond a) = P(s_0 s_1 s_3^w) = 1$ in M_1. However, $Po(s_0 \models \Diamond a) = Po(s_0 s_1 s_3^w) = 0.5$ in M_2. State s_0 satisfies $Po_{=1}(\Diamond a)$ in M_1, while s_0 does not satisfy $Po_{=1}(\Diamond a)$ in M_2. Hence, $s_0 \in Sat_{M_1}(Po_{=1}(\Diamond a))$, but $s_0 \notin Sat_{M_2}(Po_{=1}(\Diamond a))$. This implies that

$$Sat_{M_1}(Po_{=1}(\Diamond a)) \neq Sat_{M_2}(Po_{=1}(\Diamond a)). \tag{6}$$

14

Since Φ is a CTL state formulae, and $TS(M_1) = TS(M_2)$, we have

$$Sat_{TS(M_1)}(\Phi) = Sat_{TS(M_2)}(\Phi). \tag{7}$$

By the assumption $\Phi \equiv Po_{=1}(\Diamond a)$, it follows that $Sat_{TS(M)}(\Phi) = Sat_M(Po_{=1}(\Diamond a))$ for any finite possibilistic Kripke structure M. Then we have

$$Sat_{M_1}(Po_{=1}(\Diamond a)) = Sat_{M_2}(Po_{=1}(\Diamond a)). \tag{8}$$

Eq.6 and Eq.8 shows a contradiction, which proves that there is no CTL formula that is equivalent to $Po_{=1}(\Diamond a)$. □
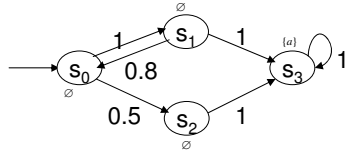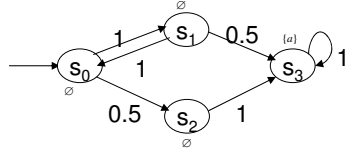

Fig.2.A finite possibilistic Kripke structure M_1.

Fig.3.A finite possibilistic Kripke structure M_2.

Combining Theorem 3.4 and Theorem 3.5, it follows that CTL is a proper subclass of PoCTL. PoCTL is completely different from probabilistic CTL. In fact, probabilistic CTL and CTL can not be comparable with each other (whereas, for finite probabilistic Kripke structure, the qualitative fragment of probabilistic CTL can be embedded into CTL and thus a proper subclass of PoCTL).

Using similar arguments, we can show that the following theorems also hold in finite possibilistic Kripke structures.

Theorem 3.6. *There is no CTL formula that is equivalent to $Po_{=1}(\bigcirc a)$.*

15

Theorem 3.7. *There is no CTL formula that is equivalent to $Po_{=1}(\Box a)$.*

Theorem 3.8. *There is no CTL formula that is equivalent to $Po_{=1}(a \sqcup b)$.*

3.3. Properties of repeated reachability and persistence

This subsection will show that qualitative properties for events such as repeated reachability - a certain set of states being visited repeated, and persistence - only a certain set of states being visited from the moment on, can be described by PoCTL formulae. And we will show that some properties that can not be expressed in CTL can be expressed in the qualitative fragment of PoCTL.

For CTL, universal repeated reachability properties ([1]) can be formalized by the combination of the modalities $\forall\Box$ and $\forall\Diamond$:

$$s \models \forall\Box\forall\Diamond a \text{ iff } \pi \models \Box\Diamond a \text{ for all } \pi \in Paths(s).$$

For finite possibilistic Kripke structures, a similar result holds for the qualitative fragment of PoCTL.

Theorem 3.9. *Let M be a finite possibilistic Kripke structure, and s a state of M. Then, we have*

$$s \models Po_{=1}(\Box Po_{=1}(\Diamond a)) \text{ iff } Po(s \models \Box\Diamond a) = 1.$$

Proof. Since $s \models Po_{=1}(\Box Po_{=1}(\Diamond a))$ if and only if $Po(s \models \Box Po_{=1}(\Diamond a)) = 1$, and $s \models \Box Po_{=1}(\Diamond a)$ iff $\pi \models \Box Po_{=1}(\Diamond a)$ for any $\pi \in Paths(s)$, it follows that $Po(s \models \Box Po_{=1}(\Diamond a)) = Po^{M_s}(\{\pi \in Paths(s) \mid \pi \models \Box Po_{=1}(\Diamond a)\}) = 1$. For any $\pi \models \Box Po_{=1}(\Diamond a)$, let $\pi = s_0 s_1 \cdots s_n \cdots$, then $Po(s_i \models \Diamond a) = 1$ for any s_i, where $i \geq 0$. It follows that $\pi \models \Box\Diamond a$. Noting that $Po^{M_s}(\pi) \leq Po^{M_s}(\{\pi' \in Paths(s) \mid \pi' \models \Box\Diamond a\}$, and thus,

$$Po(s \models \Box\Diamond a) = Po^{M_s}(\{\pi \in Paths(s) \mid \pi \models \Box\Diamond a\}) = 1.$$

Assume that $Po(s \models \Box\Diamond a) = 1$. As $Po(s \models \Box\Diamond a) = Po^{M_s}(\{\pi \in Paths(s) \mid \pi \models \Box\Diamond a\})$ and M is finite, there exists a path $\pi \models \Box\Diamond a$ satisfying $Po^{M_s}(\pi) = 1$. Let $\pi = s_0 s_1 s_2 \cdots$. Since $\pi \models \Box\Diamond a$, we have $\pi[j \cdots] \models \Diamond a$ for any $j \geq 0$, where

16

$\pi[j \cdots] = s_j s_{j+1} \cdots$. As $Po^{M_s}(\pi[j \cdots]) \geq Po^{M_s}(\pi)$ and $Po^{M_s}(\pi) = 1$, it follows that $Po^{M_s}(\pi[j \cdots]) = 1$ for any $j \geq 0$. Note that $Po^{M_s}(\pi[j \cdots]) \leq Po(s_j \models \Diamond a)$, we have $Po(s_j \models \Diamond a) = 1$ for any $j \geq 0$. Therefore, we have $Po(s_0 \models Po_{=1}(\Diamond a)) = 1$. Hence, $s \models Po_{=1}(\Box Po_{=1}(\Diamond a))$.

According to the above proof, we have:

$$s \models Po_{=1}(\Box Po_{=1}(\Diamond a)) \text{ iff } Po(s \models \Box \Diamond a) = 1.$$

$\square$

In a similar way, by the analysis of the possibility of the evens such as repeated reachability and persistence with more than 0 and equal to 1, we can show that the following theorems hold in finite possibilistic Kripke structures for atomic events.

Theorem 3.10. *Let M be a finite possibilistic Kripke structure, and s a state of M. Then, we have*

$$s \models Po_{>0}(\Box Po_{>0}(\Diamond a)) \text{ iff } Po(s \models \Box \Diamond a) > 0.$$

Recall that universal persistence properties can not be expressed in CTL ([1]). For finite possibilistic Kripke structures, PoCTL allows specifying persistence properties with possibility 1. This is stated by the following theorem.

Theorem 3.11. *Let M be a finite possibilistic Kripke structure, and s a state of M. Then, we have*

$$s \models Po_{=1}(\Diamond Po_{=1}(\Box a)) \text{ iff } Po(s \models \Diamond \Box a) = 1.$$

Theorem 3.12. *Let M be a finite possibilistic Kripke structure, and s a state of M. Then, we have*

$$s \models Po_{>0}(\Diamond Po_{>0}(\Box a)) \text{ iff } Po(s \models \Diamond \Box a) > 0.$$

3.4. Alternative way to define the equivalence between CTL and PoCTL formulae

As mentioned in Remark 4, the definition of the equivalence of PoCTL and CTL formulae is not unique. Definition 3.3 is an analogous version of the related definition of probabilistic CTL and CTL formulae. We will give another way to define the equivalence of PoCTL and CTL formulae in the following manner.

Definition 3.5. For a finite possibilistic Kripke structure $M = (S, P, I, AP, L)$ and $\alpha \in (0, 1]$, let $TS_\alpha(M) = (S, \rightarrow_\alpha, I_\alpha, AP, L)$, where $s \rightarrow_\alpha t$ iff $P(s, t) \geq \alpha$, and $s \in I_\alpha$ iff $I(s) \geq \alpha$. PoCTL formula Φ is α-equivalent to CTL formula Ψ, denoted by $\Phi \equiv_\alpha \Psi$, if $Sat_M(\Phi) = Sat_{TS_\alpha(M)}(\Psi)$ for any finite possibilistic Kripke structure M.

We shall give some properties of PoCTL using the definition of α-equivalence of PoCTL and CTL formulae for $\alpha \in (0, 1]$. The proofs are very similar to those in Section 3.2.

Proposition 3.3. *Let φ be any CTL path formula and $\alpha \in (0, 1]$. Then, we have*

$$\exists \varphi \equiv_\alpha Po_{\geq \alpha}(\varphi). \tag{9}$$

Proposition 3.4. *For any CTL formula and $\alpha \in (0, 1]$, there exists an α-equivalent PoCTL formula.*

Proposition 3.5. *For any CTL formulae Φ and Ψ, let $\alpha \in (0, 1]$, we have*
    (1) $\exists \Diamond \Phi \equiv_\alpha Po_{\geq \alpha}(\Diamond \Phi)$,
    (2) $\exists \bigcirc \Phi \equiv_\alpha Po_{\geq \alpha}(\bigcirc \Phi)$,
    (3) $\exists \Box \Phi \equiv_\alpha Po_{\geq \alpha}(\Box \Phi)$, and
    (4) $\exists (\Phi \sqcup \Psi) \equiv_\alpha Po_{\geq \alpha}(\Phi \sqcup \Psi)$.

Proposition 3.6. *For any CTL formulae Φ and Ψ, let $\alpha \in (0, 1]$, we have*
    (1) $\forall \bigcirc \Phi \equiv_\alpha Po_{< \alpha}(\bigcirc \neg \Phi)$,
    (2) $\forall (\Phi \sqcup \Psi) \equiv_\alpha Po_{< \alpha}(\neg \Psi \sqcup (\neg \Phi \wedge \neg \Psi)) \wedge Po_{< \alpha}(\Box \neg \Psi)$,
    (3) $\forall \Diamond \Phi \equiv_\alpha Po_{< \alpha}(\Box \neg \Phi)$, and
    (4) $\forall \Box \Phi \equiv_\alpha Po_{< \alpha}(\Diamond \neg \Phi)$.

18

Proposition 3.7. *For any $\alpha \in (0, 1]$, there is no CTL formula that is α-equivalent to $Po_{=1}(\Diamond a)$.*

The α-equivalence of PoCTL and CTL formulae might be useful in the approximation of PoCTL formulae using CTL formulae. This would allow a graded approach to establish a level cut to decide e.g. when a transition with value α can be considered as existing or not. The general notion of α-equivalence would be a very general approach such that the notions of equivalence (actually $> 0-$equivalence) and 1-equivalence would come out as a limit case and particular case respectively. However, intuitively, 1-equivalence is too strong to define the equivalence of PoCTL and CTL formulae in the senses as explained below. By 1-equivalence, the possibility of a certain "event" is larger than 0 does not imply that the "event" exists. For example, in Fig.3, intuitively, $s_0 \models \exists \Diamond a$. However, by a simple calculation , we have $Po(s_0 \models \Diamond a) = 0.5 < 1$. It follows that $s_0 \not\models Po_{=1}(\Diamond a)$, hence, $s_0 \not\models \exists \Diamond a$. Furthermore, intuitively, 1-equivalence is too strong for universal quantifier $\forall$. By Proposition 3.6, the universal "event" means that the possibility of the negation of the "event" is less than 1. There are "events" such that the possibility of the negation of the "events" is less than 1 but there exist some paths that violate the "events". We shall give some analysis in the illustrative example in Section 5.

4. PoCTL Model Checking

Similar to classical and probabilistic CTL model-checking problems, the PoCTL model-checking problem can be stated as follows:

For a given finite possibilistic Kripke structure M, state s in M, and PoCTL state formula Φ, decide whether $s \models \Phi$.

We write $(M, s) \models \Phi$ for this PoCTL model-checking problem.

As shown in the above section, PoCTL is more expressible than CTL. There are some PoCTL model-checking problems that can not be tackled by classical CTL model-checking algorithm. We shall present some methods to tackle PoCTL

model-checking problems in this section. The technique of PoCTL model checking is very similar to those of classical and probabilistic CTL model checking. The difference lies in the operations involving in the processing of model checking.

To determine whether $s \models \Phi$, we need to compute the satisfaction set $Sat(\Phi)$. This is done recursively using a bottom-up traversal of the parse tree of Φ with time complexity $O(|\Phi|)$, where $|\Phi|$ denotes the number of subformulae of Φ (see the definition of $|\Phi|$ in Section 6.4.3 in [1]). As for CTL model checking, the nodes of the parse tree represent the subformulae of Φ. For each node of the parse tree, which represents a subformula Ψ of Φ, the set $Sat(\Psi)$ is calculated. If Ψ is propositional logic formula, $Sat(\Psi)$ can be computed in exactly the same way as for CTL. The left part is the treatment of subformulae of the form $\Psi = Po_J(\varphi)$. Since

$$Sat(Po_J(\varphi)) = \{s \in S \mid Po(s \models \varphi) \in J\}, \tag{10}$$

to calculate $Sat(\Psi)$, we need to compute the possibility $Po(s \models \varphi)$ for any state s.

There are three ways to construct path formula φ, i.e., $\varphi = \bigcirc\Psi$, $\varphi = \Phi \sqcup^{\leq n} \Psi$ or $\varphi = \Phi \sqcup \Psi$ for some state formulae Φ and Ψ and $n \in \mathbb{N}$.

For $\varphi = \bigcirc\Psi$, the next-step operator, the following equality holds:

$$Po(s \models \bigcirc\Psi) = \bigvee_{s' \in Sat(\Psi)} P(s, s')$$

where P is the transition matrix of M. In the matrix-vector notation we thus have that the (column) vector $(Po(s \models \bigcirc\Psi))_{s \in S}$ can be computed by multiplying P with the characteristic vector for $Sat(\Psi)$, i.e., (column) bit vector $(b_s)_{s \in S}$ where $b_s = 1$ if and only if $s \in Sat(\Psi)$. Write $\chi_\Psi = (b_s)_{s \in S}$, then we have

$$(Po(s \models \bigcirc\Psi))_{s \in S} = P \circ \chi_\Psi. \tag{11}$$

It follows that, checking the next-step operator thus reduces to a single matrix-vector multiplication.

To calculate the possibility $Po(s \models \varphi)$ for until formulae $\varphi = \Phi \sqcup^{\leq n} \Psi$ or $\varphi = \Phi \sqcup \Psi$. Let $C = Sat(\Phi)$ and $B = Sat(\Psi)$, by its definition, we have

$$Po(s \models \Phi \sqcup^{\leq n} \Psi) = Po(s \models C \sqcup^{\leq n} B), \text{ and}$$

20

$$Po(s \models \Phi \sqcup \Psi) = Po(s \models C \sqcup B),$$

where $Po(s \models C \sqcup^{\leq n} B) = Po^{M_s}(\{\pi \in Paths(s) | \exists 0 \leq j \leq n, \pi[j] \in B$ and for any $0 \leq k < j, \pi(k) \in C\})$ and $Po(s \models C \sqcup B) = Po^{M_s}(\{\pi \in Paths(s) | \exists j \geq 0, \pi[j] \in B$ and for any $0 \leq k < j, \pi(k) \in C\})$

We posed a least fixed point characterization to calculate $Po(s \models C\sqcup B)$ in [19]. In the following, we shall give a direct method to calculate $Po(s \models C \sqcup^{\leq n} B)$ and $Po(s \models C\sqcup B)$, which is completely different from the method used in probabilistic CTL model checking for until operator, where a linear equation system needs to be solved with more time complexity.

As done in [19], let $S_{=0}, S_{=1}, S_?$ be a partition of S such that,

(1) $B \subseteq S_{=1} \subseteq \{s \in S | Po(s \models C \sqcup B) = 1\}$;

(2) $S \backslash (C \cup B) \subseteq S_{=0} \subseteq \{s \in S | Po(s \models C \sqcup B) = 0\}$;

(3) $S_? = S \backslash (S_{=1} \cup S_{=0})$.

The above partition of S always exists. For example, we can take $S_{=1} = B$, $S_{=0} = S \backslash (C \cup B)$ and $S_? = S \backslash (S_{=1} \cup S_{=0}) = C - B$. Note that the technique and notations used here have been adopted from probabilistic CTL model checking [1].

For all state s, write

$$x_s = Po(s \models C \sqcup^{\leq n} B).$$

If $s \in S_{=1}$, we have $Po(s \models C\sqcup^{\leq n} B) = 1$; if $s \in S_{=0}$, $Po(s \models C\sqcup^{\leq n} B) = 0$; if $s \in S_?$, we can get a fuzzy matrix $P_? = (P_?(s,t))_{s,t \in S}$ by letting $P_?(s,t) = P(s,t)$ whenever $s, t \in S_?$ and 0 otherwise. The left is to give a method to calculate $(x_s)_{s \in S_?}$.

By the definition of $C \sqcup^{\leq n} B$, we have

$$\{\pi \in Paths(s) | \pi \models C \sqcup^{\leq n} B\}$$
$$= \{\pi \in Paths(s) | \exists k \leq n, \text{ if } 0 \leq i < k, \pi(i) \in C, \text{ and } \pi(k) \in B\}$$
$$= \bigcup \{Cyl(s_0 \cdots s_k t) | 0 \leq k \leq n, s_0 = s, s_1, \cdots, s_k \in C \text{ and } t \in B\}.$$

21

Hence,

$$Po(s \models C \sqcup^{\leq n} B)$$

$$= \bigvee_{k=0}^{n} \bigvee \{Po(Cyl(s_0 \cdots s_k t))|s_0 = s, s_1, \cdots, s_k \in C \text{ and } t \in B\}.$$

Write $\chi_s = (a_t)_{t \in S_?}$ for the (row) characteristic vector for the singleton $\{s\}$, i.e., $a_t = 1$ if $t = s$ and $a_t = 0$ if $t \neq s$; $\chi_B = (b_t)_{t \in S_?}$ for the (column) characteristic vector for B, i.e., $b_t = 1$ if $t \in B$ and 0 otherwise. By a simple calculation, we have

$$\bigvee \{Po(Cyl(s_0 \cdots s_k t))|s_0 = s, s_1, \cdots, s_k \in C \text{ and } t \in B\} = \chi_s \circ P_?^k \circ P \circ \chi_B$$

for any k. It follows that

$$x_s = Po(s \models C \sqcup^{\leq n} B) = \bigvee_{k=0}^{n} \chi_s \circ P_?^k \circ P \circ \chi_B = \chi_s \circ \bigvee_{k=0}^{n} P_?^k \circ P \circ \chi_B.$$

If we write $P_?^{\leq n} = \bigvee_{k=0}^{n} P_?^k$, where $P_?^0$ is the identity matrix, i.e., $P_?^0(s, s) = 1$ and 0 otherwise, then

$$x_s = Po(s \models C \sqcup^{\leq n} B) = \chi_s \circ P_?^{\leq n} \circ P \circ \chi_B.$$

Hence, if we write $\chi_? = (\chi_?(s, t))_{s \in S_?, t \in S}$ as the characteristic matrix for $S_?$ in S, i.e., $\chi_?(s, s) = 1$ for $s \in S_?$ and 0 otherwise, then we have

$$(x_s)_{s \in S_?} = \chi_? \circ P_?^{\leq n} \circ P \circ \chi_B. \tag{12}$$

To calculate $(x_s)_{s \in S_?}$, it is sufficient to perform matrix multiplication at most $n + 3$ times. Observe that, if $n \geq |S_?|$, then $P_?^{\leq n} = P_?^0 \vee P_?^+$, which is denoted by $P_?^*$. Then $P_?^*$ is the reflexive and transitive closure of the fuzzy matrix $P_?$. In this case, we have

$$(x_s)_{s \in S_?} = \chi_? \circ P_?^* \circ P \circ \chi_B. \tag{13}$$

In particular, we have

$$(x_s)_{s \in S_?} = (Po(s \models C \sqcup B))_{s \in S_?} = \chi_? \circ P_?^* \circ P \circ \chi_B. \tag{14}$$

22

In summary, we have

$$x_s = Po(s \models C \sqcup^{\leq n} B) = \begin{cases} 1, & \text{if } s \in S_{=1}, \\ 0, & \text{if } s \in S_{=0}, \\ \chi_s \circ P_?^{\leq n} \circ P \circ \chi_B, & \text{if } s \in S_?. \end{cases} \quad (15)$$

In particular, if $n \geq |S_?|$, we have

$$x_s = Po(s \models C \sqcup^{\leq n} B) = Po(s \models C \sqcup B) = \begin{cases} 1, & \text{if } s \in S_{=1}, \\ 0, & \text{if } s \in S_{=0}, \\ \chi_s \circ P_?^* \circ P \circ \chi_B, & \text{if } s \in S_?. \end{cases} \quad (16)$$

In the calculation of $(x_s)_{s \in S}$, we only need to perform (fuzzy) matrix multiplication at most $N(= |S|) + 3$ times. It follows that the time complexity of PoCTL model checking of a finite possibilistic Kripke structure M and a PoCTL formula Φ can be presented as follows.

Theorem 4.1. *(Time Complexity of PoCTL Model Checking) For a finite possibilistic Kripke structure M, state s in M, and a PoCTL formula Φ, the PoCTL model-checking problem $(M, s) \models \Phi$ can be determined in time $O(size(M) \cdot N \cdot |\Phi|)$, where $|\Phi|$ denotes the number of subformulae of Φ.*

5. An illustrative example

We now give an example to illustrate the PoCTL model-checking approach presented in this paper. The same example is used in [19] to illustrate the application of model checking of linear-time properties based on possibility measures. Note that this is a demonstrative rather than a case study aimed at showing the scalability of our approach.

Suppose that there is an animal with a new disease. For the new disease, the doctor has no complete knowledge about it, but he (or she) believes by experience that the drug Ribavirin may be useful for the treating the disease.

For simplicity, it is assumed that the doctor considers roughly the animal's condition to be three states, say, "poor", "fair" and "excellent". It is vague when the animal's condition is said to be "poor", "fair" and "excellent". The doctor will use the fuzzy set (called fuzzy state in the following) over states "poor",

23

"fair" and "excellent" to describe the animal condition (see [4, 20, 21] for more explanations). Therefore, when a possibilistic Kripke structure is used to model the treatment processes of the animal, a fuzzy state is naturally denoted as a three-dimensional vector $[a_1, a_2, a_3]$, which is represented as the possibility distribution of the animal's condition over states "poor", "fair" and "excellent".

Similarly, it is imprecise to say that at what point exactly the animal has changed from one state to another state after a drug treatment (i.e., event), because the drug event occurring may lead a state to fuzzy state "poor", "fair" and "excellent". Therefore, the treatment process is modeled by a possibilistic Kripke structure, in which a transition possibility distribution is represented by a 3×3 matrix.

Suppose that the treatment process of the animal is modeled by the following possibilistic Kripke structure $M = (S, P, I, AP, L)$, where $S = AP = \{poor, fair, excellent\}$,

$$P = \begin{pmatrix} 0.2 & 1 & 1 \\ 0.2 & 0.5 & 1 \\ 0.5 & 1 & 0.5 \end{pmatrix}, I = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

and $L(s) = \{s\}$ for any $s \in S$.

The structure M is presented in Fig.4, and the corresponding M^+ is presented in Fig.5, where we use the symbols p, f, e to represent the states or the atomic propositions "poor", "fair" and "excellent" respectively.
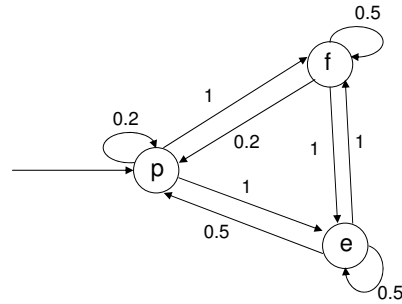

Fig.4.The possibilistic Kripke structure M for the treatment process of the animal.
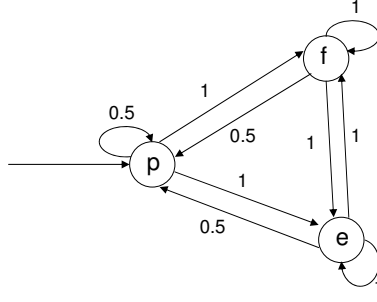
Fig.5.The corresponding M^+ of M in Fig.4.

By a simple calculation, we have

$$P^+ = \begin{pmatrix} 0.5 & 1 & 1 \\ 0.5 & 1 & 1 \\ 0.5 & 1 & 1 \end{pmatrix}.$$

Some calculations are presented as follows in detail.

(1) Let us calculate $Po(poor \models \{poor\} \sqcup^{\leq 7} \{excellent\})$. In this case, let us take $S_{=1} = \{excellent\}$, $S_{=0} = \{fair\}$, and $S_? = \{poor\}$. It follows that, $P_? = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, and then $P_?^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. By Eq.16, we have

$$Po(poor \models \{poor\} \sqcup^{\leq 7} \{excellent\})$$

$$= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \circ P_?^{\leq 7} \circ P \circ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \circ P_?^* \circ P \circ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= 1.$$

Hence, $poor \models Po_{=1}(\{poor\} \sqcup^{\leq 7} \{excellent\})$. It means that the animal will be recovered after one week treatment with possibility 1.

25

(2) Since

$$Po(poor \models \Diamond\{excellent\}) = Po(poor \models true \sqcup \{excellent\})$$

$$= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \circ \begin{pmatrix} 1 & 0.5 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 0.2 & 1 & 1 \\ 0.2 & 0.5 & 1 \\ 0.5 & 1 & 0.5 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= 1.$$

In this case, we take $S_{=1} = \{excellent\}$, $S_{=0} = \emptyset$ and $S_? = \{poor, fair\}$.

Hence, $poor \models Po_{=1}(\Diamond\{excellent\})$.

(3) We have $poor \not\models \forall\Diamond\{excellent\}$. The reason is as follows. By Proposition 3.2(3), we have

$$\forall\Diamond\{excellent\} \equiv Po_{=0}(\Box\neg\{excellent\}).$$

Let us calculate $Po(s \models \Box\neg\{excellent\})$, where $s = poor$:

$$Po(s \models \Box\neg\{excellent\}) = Po^{M_s}(\{\pi \in paths(s)|\pi \models \Box\neg\{excellent\}) = Po^{M_s}(pf^\omega) = 0.5 > 0.$$

Hence, $s = poor \not\models Po_{=0}(\Box\neg\{excellent\})$, i.e., $poor \not\models \forall\Diamond\{excellent\}$.

Since $\forall\Diamond\{excellent\} \equiv_1 Po_{<1}(\Box\neg\{excellent\})$, and $Po(poor \models \Box\neg\{excellent\}) = 0.5 < 1$, it follows that $poor \models_1 \forall\Diamond\{excellent\}$ if we adopt 1-equivalence. This is too strong, since we still have the event p^ω (with possibility 0.2) and the event pf^ω (with possibility 0.5), and the above two events (may occur) violate the property $\forall\Diamond\{excellent\}$.

(2) and (3) show that $\forall\Diamond\Phi \equiv Po_{=1}(\Diamond\Phi)$ does not hold in PoCTL.

(4) Let $s = poor$, $a = excellent$, by Theorem 3.9, we have

$$s \models Po_{=1}(\Box Po_{=1}(\Diamond a)) \text{ iff } Po(s \models \Box\Diamond a) = 1.$$

It has been shown that $Po(s \models \Box\Diamond a) = 1$ in [19]. Then we know that

$$s \models Po_{=1}(\Box Po_{=1}(\Diamond a)).$$

(5) Since $Po(s \models \Box\neg\{poor\}) = 0$, where $s = poor$. It follows that $poor \models \forall\Diamond\{poor\}$.

26

6. Conclusion

This paper is a continuation of previous work in the papers [19, 26], where LTL model checking based on possibility measures and possibilistic CTL were introduced. We further studied the expressiveness of PoCTL and PoCTL model cheking in this paper, which was not considered in [19, 26]. The main contribution of this paper is as follows. We showed that (qualitative) PoCTL is more powerful than CTL with respect to their expressiveness. In particular, we have shown that any CTL formula is equivalent to a qualitative PoCTL formula. Some basic PoCTL formulae that are not equivalent to any CTL formulae were also given. Some qualitative repeated reachability and persistence properties were expressed using PoCTL formulae. The PoCTL model checking problem was discussed in detail. The method of PoCTL model checking were given and its time complexity was analyzed.

This is the first step of PoCTL model checking. There are many things that can be done based on this.

As we know, there are many industrial model checkers related to CTL model checking, including SMV ([25]) and NuSMV. Since CTL is a proper subclass of PoCTL, it is necessary to set up some model checker corresponding to PoCTL model checking. The equivalence and abstraction technique corresponding to PoCTL model checking are also necessary to be investigated in the future work.

Of course, the research directions related to possibilistic LTL model checking posed in [19] can also be applied to PoCTL model checking. We list three of them as follows.

- We use max-min composition of fuzzy relations in this paper. There are other forms of composition of fuzzy relations, such as max-product composition which are useful for the applications of fuzzy sets. Then the related work using other composition instead of max-min composition can be done in the future.

27

- We use the normal possibility distributions in this paper (see conditions (2) and (3) in the definition of possibilistic Kripke structure). How to deal with those possibilistic Kripke structures which do not satisfy conditions (2) and (3) is another future direction to study.

- In the definition of possibilistic Kripke structures, the labeling function $L : S \to 2^{AP}$ is crisp, there is no vagueness at all here. This restriction is too strict. How to dealt with the possibilistic Kripke structures with uncertainty labeling function in PoCTL is still another issue needed to be discussed further. Although we can transform a possibilistic Kripke structure with uncertainty labeling function into a possibilistic Kripke structure with classical labeling function as noted in [19], a direct method using possibilistic Kripke structures with uncertainty labeling functions still deserves study.

Acknowledgments

References

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

[2] P. Baltazar, P. Mateus, R. Nagarajan, N. Papanikolaou, *Exogenous probabilistic computation tree logic*, Electronic Notes in Theoretical Computer Science, 190(2007) 95-110.

[3] R. Barbuti, F. Levi, P. Milazzo, G. Scatena, *Probabilistic model checking of biological systems with uncertain kinetic rates*, Theoretical Computer Science, 419(2012) 2-16.

[4] Y.Cao, M. Ying, *Observability and decentralized control of fuzzy discrete-event systems*, IEEE Transactions on Fuzzy Systems,14(2)(2006) 202-216.

[5] F.Ciesinski, M.Grober, *On probabilistic computation tree logic*, Lecture Notes in Computer Science, 2925(2004) 333-355.

[6] E.Clarke, E. Emerson, A.Sistla, *Automatic verification of finite-State concurrent systems using temporal logic specifications*, ACM Transactions on Programming Languages and Systems, 8(2)(1986) 244-263.

[7] L. Cui, Y. Li, *Linguistic quantifiers based on Choquet integrals*, International Journal of Approximate Reasoning, 48(2008) 559-582.

[8] A. Dovier, E. Quintarelli, *Applying model-checking to solve queries on semistructured data*, Computer Languages, Systems and Structures, 35(2009) 143-172.

[9] A. Drakopoulos, *Probabilities, possibilities, and fuzzy sets*, Fuzzy Sets and Systems,75(1995) 1-15.

[10] D. Dubois , F. Dupin de Saint Cyr, H. Prade, *Updating, transition constraints and possibilistic Markov chains*, Proc. of IPMU 1994, 826-831, 1994.

[11] D. Dubois, H. Prade, *Possibility theory, probability theory and multiple-valued logics: A clarification*, Annals of Mathematics and Artificial Intelligence, 32(2001) 35-66.

[12] M. Edmund, O. Grumberg, D. Peled, *Model Checking*, Cambridge: the MIT Press, 1999.

[13] M. Grabisch, T. Murofushi, M. Sugeno (eds), *Fuzzy Measures and Integrals*, Heidelberg New Tork: Physica-Verlag, 2000.

[14] P. Hájek, D. Harmancová, R. Verbrugge, *A qualitative fuzzy possibilistic logic*, International Journal of Approximate Reasoning, 12(1995) 1-19.

[15] S. Hart, M. Sharir, *Probabilistic propositional temporal logics*, Information and Control,70(2-3)(1986) 97-155.

[16] H. Janssen, G. de Cooman and E.E. Kerre, *First results for a mathematical theory of possibilistic Markov processes*, in: Proceedings of IPMU'96, Vol. III (Information Processing and Management of Uncertainty in Knowledge-Based Systems), Granada, Spain (1996) pp. 1425- 1431.

[17] R. Kruse, R. Buck-Emden, R. Cordes, *Processor power considerations - An application of fuzzy Markov chains*, Fuzzy Sets and Systems, 21(1987) 289-299.

[18] Y. Li, *Analysis of Fuzzy Systems(in Chinese)*, Beijing, China: Science Press, 2005.

[19] Y. Li, L. Li, *Model checking of linear-time properties based on possibility measure*, IEEE Transactions on Fuzzy Systems, 21(5)(2013), 842-854.

[20] F. Lin, H. Ying, *Modeling and control of fuzzy discrete event systems*, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 32(4)(2002) 408-415.

[21] F. Liu, D. Qiu, *Diagnosability of fuzzy discrete-event systems: a fuzzy approach*, IEEE Transactions on Fuzzy Systems, 17(2)(2009) 372-384.

[22] A.C.Patthak, I.Bhattacharya, A.Dasgupta, Pallab Dasgupta, P.P.Chakrabarti, *Quantified computation tree logic*, Information Processing Letters, 8(2002) 123-129.

[23] M. Sugeno, *Theory of Fuzzy Integrals and its Applications*,   PhD thesis, Tokyo Institute of Technology, 1974.

[24] K.Y.Rozier, *Linear temporal logic symbolic model checking*, Computer Science Review, 5(2011) 163-203.

[25] K. McMillan, *Symbolic Model Checking*, Dordrecht, The Netherland: Kluwer, 1993.

[26] Y. Xue, H. Lei, Y. Li, *Computationg tree logic based on possibility measure(in Chinese)*, Computer Engineering and Science,33(9)(2011) 70-75.

[27] M. Ying, *Linguistic quantifiers modeled by Sugeno integrals*, Artificial Intelligence, 170(6-7)(2006) 581-606.

[28] L.A. Zadeh, *Fuzzy sets*, Information and Control, 8(1965) 338-353.

[29] L.A. Zadeh, *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems, 1 (1978) 3-28.