

Article

The μ -Calculus Model-Checking Algorithm for Generalized Possibilistic Decision Process

Jiulei Jiang ^{1,2,*}, Panqing Zhang ¹ and Zhanyou Ma ²

¹ School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; 1995012@nun.edu.cn

² Ningxia Key Laboratory of Intelligent Information and Big Data Processing, Yinchuan 750021, China; mazhany@nmu.edu.cn

* Correspondence: 1994029@nun.edu.cn

Received: 23 February 2020; Accepted: 7 April 2020; Published: 9 April 2020



Abstract: Model checking is a formal automatic verification technology for complex concurrent systems. It is used widely in the verification and analysis of computer software and hardware systems, communication protocols, security protocols, etc. The generalized possibilistic μ -calculus model-checking algorithm for decision processes is studied to solve the formal verification problem of concurrent systems with nondeterministic information and incomplete information on the basis of possibility theory. Firstly, the generalized possibilistic decision process is introduced as the system model. Then, the classical proposition μ -calculus is improved and extended, and the concept of generalized possibilistic μ -calculus ($GP\mathcal{O}_\mu$) is given to describe the attribute characteristics of nondeterministic systems. Then, the $GP\mathcal{O}_\mu$ model-checking algorithm is proposed, and the model-checking problem is simplified to fuzzy matrix operations. Finally, a specific example and a case study are analyzed and verified. Compared with the classical μ -calculus, the generalized possibilistic μ -calculus has a stronger expressive power and can better characterize the attributes of nondeterministic systems. The model-checking algorithm can give the possibility that the system satisfies the attributes. The research work provides a new idea and method for model checking nondeterministic systems.

Keywords: concurrent systems; generalized possibilistic decision process; generalized possibilistic μ -calculus; model checking; fuzzy matrix; verification problem

1. Introduction

The continuous enhancement of computer functions makes systems increasingly complex; for the purpose of the correctness of the systems, it usually spends more time on verification than on construction [1]. Formal methods are important methods to ensure the correctness and security of computer systems. They can integrate verification early in the design process and ensure the credibility of the system through strict logical reasoning and mathematical calculations.

Model checking is a formal automatic verification technology that can provide a complete formal verification framework of system attributes [2]. The classical model-checking technology is mainly used for qualitative research on the system—that is, to verify if the system satisfies the system attributes described by temporal logic formulas, such as Computation Tree Logic (CTL), Linear Temporal Logic (LTL), and μ -calculus [3,4]. If it is satisfied, the model checker returns “True”; otherwise, it returns “False” with specific counterexamples. However, in the design of actual systems, nondeterministic systems containing uncertain or inconsistent information are often encountered, so the qualitative research on the systems cannot meet the actual needs. In recent years, some scholars

have proposed quantitative extensions to classical model checking, such as models that embed features into probability [5,6], possibility [7–9], and multi-valued [10–12], etc.

Different model-checking approaches are applicable to different model types. Narasimha et al. [13] proposed a model-checking algorithm based on the probabilistic labeled transition systems and μ -calculus to check whether the states in the finite probabilistic labeled transition systems satisfy the logical formulas; Chechik et al. [14] extended the classical CTL and Kripke structure and proposed a multi-valued model checking algorithm. Gurfinkel et al. [15] studied the multi-valued μ -calculus model checking problem based on multi-valued Kripke structures and reduced it into several classical model-checking problems. The advantage of the reduction method is that the verification can be done automatically using existing model-checking tools. Mallya et al. [16] defined a multi-valued μ -calculus and proposed a new model-checking logic framework to verify arbitrary properties of multi-valued μ -calculus, which is more widely used.

Recently, Pan et al. [17] combined fuzzy logic with CTL, proposed Fuzzy Computation Tree Logic (FCTL), which is a fuzzy extension of classical CTL, and discussed model-checking problems. Li et al. [18–21] extended the classical LTL and CTL model-checking technology; they defined a quantitative model-checking verification method on the basis of possibility measures. Compared to probabilistic model checking, the possibilistic model checking does not need to satisfy countable additivity, and it is mainly used for the model checking of non-additive systems. Probabilistic model checking already has mature model-checking tools, such as PRISM PRISM is a probabilistic model checker, a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour. It has been used to analyse systems from many different application domains, including communication and multimedia protocols, randomised distributed algorithms, security protocols, biological systems and many others [22], etc., while possibilistic model checking is still in the theoretical research. Possibilistic model checking uses the theory of possibility measure as the theoretical basis for system modeling, which is suitable for systems with uncertain information in possibility theory [7].

There is some research on quantitative model-checking verification methods on the basis of possibility measures. Li et al. proposed Generalized Possibilistic LTL (GPoLTL), which is an extension of LTL, and gave quantitative model checking methods of linear-time properties based on generalized possibility measures in [18]. They also extended CTL to Generalized Possibilistic CTL (GPoCTL) and proposed a model-checking algorithm under the generalized possibilistic decision process in [21]. Our paper is the first to extend classical μ -calculus in possibility measure theory, and it studies the possibilistic model checking. The μ -calculus is very expressive and it can capture many other temporal (such as CTL*) and program logics [12]. We extend classical μ -calculus by adding a possibility value, which is denoted as Generalized Possibilistic μ -calculus (GPo μ); the semantics interpret the GPo μ formulas as mappings from the set of states of Generalized Possibilistic Decision Process (GPDP) to the domain of (0,1). The conjunction, disjunction, and negation logical operators are interpreted as the meet, join, and complementation operators in (0,1), respectively, so we use the fuzzy logic and possibility measure theory for reasoning and calculus attribute values of the systems. The GPo μ can express some properties that GPoCTL and GPoLTL cannot describe. Finally, we solve the model-checking problems by fuzzy matrices.

This paper is organized as follows. Section 2 gives basic knowledge of fuzzy theory and possibility measure theory. In Section 3, the notion of the generalized possibilistic decision process is introduced as the model of nondeterministic systems. Section 4 introduces the generalized possibilistic μ -calculus (GPo μ) to characterize the attributes for uncertain systems and gives a model-checking algorithm to verify the possibilities that the system states satisfy the attributes. A specific example and case study are explained in Section 5. We give a conclusion at the end of the paper.

2. Fuzzy Theory and Possibility Measure Theory

2.1. Fuzzy Theory

The *fuzzy set* [23] on domain U is defined by a function $A : U \rightarrow [0, 1]$; then, A is called a *fuzzy subset* on U . If the element in the domain U is represented by x , and $A(x) \in [0, 1]$, then $A(x)$ is the *membership degree* that x belongs to A .

Let A and B be two fuzzy subsets on U . For any $x \in U$, the membership functions of *union* $A \cup B$, *intersection* $A \cap B$ *complement* A^c , and *implication* $A \rightarrow B$, are defined as follows:

$$(A \cup B)(x) = A(x) \vee B(x) = \max\{A(x), B(x)\}$$

$$(A \cap B)(x) = A(x) \wedge B(x) = \min\{A(x), B(x)\}$$

$$A^c(x) = 1 - A(x)$$

$$(A \rightarrow B) = A(x) \rightarrow B(x) = A^c(x) \vee B(x)$$

Let $Q = (q_{ij})_{m \times n}$ be a matrix; if $(q_{ij}) \in [0, 1]$, then Q is a fuzzy matrix. For arbitrary fuzzy matrices $Q, R, Q = (q_{ij})_{m \times n}, R = (r_{ij})_{m \times n}$, the union of the fuzzy matrix $Q \cup R$, intersection $Q \cap R$, and the complement operation Q^c are defined as follows:

$$Q \cup R = (q_{ij} \vee r_{ij})_{m \times n}$$

$$Q \cap R = (q_{ij} \wedge r_{ij})_{m \times n}$$

$$Q^c = (1 - q_{ij})_{m \times n}$$

Given fuzzy matrices $Q = (q_{ik})_{m \times p}, r = (r_{kj})_{p \times n}$, their $\vee - \wedge$ composite operation, denoted by $Q \circ R$, which is defined by $Q \circ R = \bigvee_{k=1}^p (q_{ik} \wedge r_{kj})$.

2.2. Possibility Measure Theory

Let I be the index set, the possibility measure on nonempty set U is a mapping $POS: 2^U \rightarrow [0, 1]$, satisfying the following formulae [21]:

$$POS(\emptyset) = 0, (2) POS(U) = 1, (3) \text{ If } E_i \subseteq U, \text{ and } i \in I, \text{ then } POS(\cup_{i \in I} E_i) = \vee_{i \in I} POS(E_i).$$

If POS only satisfies the Formulae (1) and (3), then POS is called a *generalized possibility measure*. If POS is a generalized possibility measure on nonempty set U , for any $E \subseteq U$, satisfy $POS(E) = \vee_{x \in E} POS(\{x\})$.

3. Generalized Possibilistic Decision Process

In the design of practical systems, there are systems that have both nondeterministic choices of actions and possibility distributions of states. We give the notion of the generalized possibilistic decision process as the models for uncertainty systems.

Definition 1. A *generalized possibilistic decision process (GPDP)* [24] is a tuple $M = (S, I, Act, R, AP, L)$, where S is a finite, nonempty set of states; $I : S \rightarrow [0, 1]$ is the *possibilistic initial distribution function*; Act is a set of actions; $R : S \times Act \times S \rightarrow [0, 1]$ is the *possibilistic transition distribution function*, and for every state $s \in S$ and every action $\alpha \in Act$, there is a state s' such that $R(s, \alpha, s') > 0$; AP is a set of atomic propositions; $L : S \times AP \rightarrow [0, 1]$ is a *labeling function*, $L(s, a)$ denotes the possibility that the atomic proposition a holds on state s .

If S and AP are both finite, we call M a finite GPDP. $R(s, \alpha, t)$ represents the possibility that the system evolves from state s into state t by action α . If $R(s, \alpha, t) > 0$, state t is called the successor of state s , and state s is the predecessor of state t . The set of direct α -successors of s is defined as: $Post(s, \alpha) = \{t \in S | R(s, \alpha, t) > 0\}$. The set of predecessors of t is defined by: $Pre(t) = \{s \in S | R(s, \alpha, t) > 0\}$. The set of actions that state s can trigger is defined as: $Act(s) = \{\alpha \in Act | \exists s' \in S, R(s, \alpha, s') > 0\}$.

The paths in the GPDP M is denoted as π , and $\pi = s_0\alpha_0s_1\alpha_1s_2 \dots$. $Paths(s)$ and $Paths_{fin}(s)$ denote the set of infinite paths and the set of finite paths starting from state s , respectively. $Paths(M)$ and $Paths_{fin}(M)$ represent the set of all infinite paths and the set of all finite paths in M , respectively.

In $M = (S, I, Act, R, AP, L)$, for any $\alpha \in Act$, we can use the fuzzy matrix to express the possibilistic transition distribution function $R : S \times \alpha \times S \rightarrow [0, 1]$, which is denoted as $R_\alpha, R_\alpha = (R(s, \alpha, s'))_{s, s' \in S}$, in which R_α is the fuzzy transition matrix corresponding to action α in M . The maximum possibilistic transition matrix is expressed as $R_{max} = \bigvee_{i=0}^n R_{\alpha_i}$, which is also expressed by

$$(R_{max}(s, t))_{s, t \in S} = \left(\bigvee_{\alpha \in Act(s)} R(s, \alpha, t) \right)_{s, t \in S}.$$

The minimum possibilistic transition matrix is expressed as $R_{min} = \bigwedge_{i=0}^n R_{\alpha_i}$, that is,

$$(R_{min}(s, t))_{s, t \in S} = \left(\bigwedge_{\alpha \in Act(s)} R(s, \alpha, t) \right)_{s, t \in S}.$$

The transition matrix closure of a fuzzy matrix R is denoted as $R^+, R^+ = R \vee R^2 \vee R^3 \vee \dots \vee R^{|S|}$, where $R^{k+1} = R^k \circ R$. The reflexive transition closure of the fuzzy matrix R is denoted as $R^*, R^* = R^0 \vee R^+$, where R^0 denotes the identity matrix.

Let us take the generalized possibilistic decision process (GPDP) in Figure 1 as an example. The set of system states is denoted as $S = \{s_0, s_1, s_2, s_3\}$; $I = \{s_0\}$ represents the initial state of the system; $Act = \{\alpha, \beta\}$ denotes the set of actions; $R(s_0, \beta, s_1) = 0.72$ denotes that the possibility of the system changing from state s_0 to s_1 by action β is 0.72; the set of atomic propositions is $AP = \{a, b, c\}$. $L(s_0, b) = 0.4$ denotes that the possibility that atomic proposition b holds in the state s_0 is 0.4.

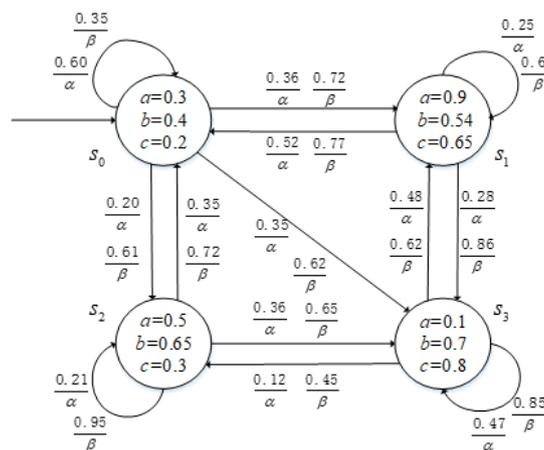


Figure 1. Generalized possibilistic decision process (GPDP).

The corresponding fuzzy matrices of the generalized possibilistic decision process in Figure 1 is given in the order $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$:

$$\begin{aligned}
 R_\alpha &= \begin{pmatrix} 0.8 & 0.3 & 0.8 \\ 0.1 & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.1 \end{pmatrix} & R_\beta &= \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.8 \\ 0.5 & 0.7 & 0.8 \end{pmatrix} & R_\gamma &= \begin{pmatrix} 0.2 & 0.7 & 0.2 \\ 0.6 & 0.5 & 0.9 \\ 0.8 & 0.5 & 0.9 \end{pmatrix} \\
 R_{max} &= \begin{pmatrix} 0.8 & 0.7 & 0.8 \\ 0.6 & 0.5 & 0.9 \\ 0.8 & 0.7 & 0.9 \end{pmatrix} & R_{min} &= \begin{pmatrix} 0.2 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.1 \end{pmatrix}
 \end{aligned}$$

Definition 2. For a generalized possibilistic decision process $M = (S, I, Act, R, AP, L)$, a function $GPO^M: Paths(M) \rightarrow [0,1]$ is defined as follows:

$$GPO^M(\pi) = I(s_0) \wedge \bigwedge_{i \geq 0} R(s_i, \alpha_i, s_{i+1}),$$

where $\pi = s_0 \alpha_0 s_1 \alpha_1 s_2 \dots \in Paths(M)$. Furthermore, for any $E \subseteq paths(M)$, we define

$$GPO(E) = \vee \{GPO(\pi) \mid \pi \in E\}.$$

Hence, the function $GPO: 2^{Paths(M)} \rightarrow [0,1]$ is a generalized possibility measure on $\Omega = 2^{Paths(M)}$.

For a GPDP $M = (S, I, Act, R, AP, L)$, the function $r : S \rightarrow [0, 1]$ is defines as:

$$r(s) = \vee \{ \bigwedge_{i \geq 0} R(s_i, \alpha_i, s_{i+1}) \mid s_1 = s, s_i \in S, \alpha_i \in Act \},$$

then $r(s)$ denotes the maximum possibility measure of a path starting from state s .

Furthermore, any finite Generalized Possibilistic Kripke Structure (GPKS) [18] can be regarded as a finite GPDP with only one action available in any state.

In the generalized possibilistic decision process, once the possibility distribution is selected by actions indeterminately, the selection of the next state is also performed by possibility selection. The uncertainty process describes the alternation of concurrent processes in a distributed system with incomplete information. Therefore, the generalized possibilistic decision process is very suitable as a model for uncertain systems.

4. GPO_μ Model-Checking Algorithm

4.1. Generalized Possibilistic μ -Calculus

The classical μ -calculus [2] is used to represent the properties of the transition systems. It is mainly used for qualitative research on the Boolean systems, but it has certain limitations; it can not describe the properties of nondeterministic systems with possibility information. In this section, we extend the classical μ -calculus by adding the possibility value, and propose the concept of generalized possibilistic μ -calculus (GPO_μ); the semantics interpret the GPO_μ formulas as mappings from the set of states of GPDP to the domain of $[0,1]$. The conjunction, disjunction, and negation logical operators are interpreted by fuzzy theory and possibility measure theory, so as to analyze the safety and reliability of uncertain systems.

Definition 3. (Syntax of GPO_μ) Let AP be the set of atomic propositions, Φ, Ψ be GPO_μ formulae, and $Var = \{X_1, X_2, X_3 \dots\}$ be a set of relational variables; the generalized possibilistic μ -calculus is defined recursively by the grammar:

$$\Phi := p \mid X \mid j \mid \neg \Phi \mid \Phi \vee \Psi \mid \Phi \wedge \Psi \mid \diamond \Phi \mid \square \Phi \mid \mu X. \Phi(X) \mid \nu X. \Phi(X)$$

where $p \in AP, X \in Var, j \in [0, 1]$.

Here, to ensure the fixpoint formulas is monotonic, the variable X should be under an even number of negations. The following are the formulas of negativity and duality under the generalized possibilistic μ -calculus:

$$\neg \neg \Phi \equiv \Phi$$

$$\begin{aligned} \neg(\Phi \wedge \Psi) &\equiv \neg\Phi \vee \neg\Psi \\ \neg\mu X.\Phi(X) &\equiv \nu X.\neg\Phi(\neg X) \\ \neg\nu X.\Phi(X) &\equiv \mu X.\neg\Phi(\neg X). \end{aligned}$$

Definition 4. (Semantics of GPO_μ) Given a generalized possibilistic decision process (GPDP) and GPO_μ formulas, Var denotes a set of relational variables, $s \in S$ is a state. The environment is defined as $\varepsilon : Var \rightarrow j^S$, in which j^S denotes the mapping from state to possibility value. For GPO_μ formula, its semantic is a fuzzy set $\|\Phi\|_\varepsilon : S \rightarrow [0, 1]$, which is defined as follows:

$$\|p\|_\varepsilon(s) = L(s, p) \tag{1}$$

$$\|X\|_\varepsilon(s) = \varepsilon(X) \tag{2}$$

$$\|j\|_\varepsilon(s) = j \tag{3}$$

$$\|\neg\Phi\|_\varepsilon(s) = 1 - \|\Phi\|_\varepsilon(s) \tag{4}$$

$$\|\Phi \vee \Psi\|_\varepsilon(s) = \|\Phi\|_\varepsilon(s) \vee \|\Psi\|_\varepsilon(s) \tag{5}$$

$$\|\Phi \wedge \Psi\|_\varepsilon(s) = \|\Phi\|_\varepsilon(s) \wedge \|\Psi\|_\varepsilon(s) \tag{6}$$

$$\|\diamond \Phi\|_\varepsilon(s) = \vee_{t \in S, \alpha \in Act} (R(s, \alpha, t) \wedge \|\Phi\|_\varepsilon(t)) \tag{7}$$

$$\|\square \Phi\|_\varepsilon(s) = \wedge_{t \in S, \alpha \in Act} (R(s, \alpha, t) \rightarrow \|\Phi\|_\varepsilon(t)) \tag{8}$$

$$\|\mu X.\Phi(X)\|_\varepsilon(s) = \wedge \{Z \in j^S \mid \|\Phi\|_{\varepsilon[X \leftarrow Z]} \leq Z\}(s) \tag{9}$$

$$\|\nu X.\Phi(X)\|_\varepsilon(s) = \vee \{Z \in j^S \mid Z \leq \|\Phi\|_{\varepsilon[X \leftarrow Z]}\}(s) \tag{10}$$

Furthermore, $\varepsilon[X \leftarrow Z]$ indicates a new environment, except that for $\varepsilon[X \leftarrow Z](X) = Z$, this environment is exactly the same as that of ε .

Theorem 1. Let Φ be a GPO_μ formula; for any $s \in S$, it satisfies the following equations.

$$\neg\|\square \Phi\|_\varepsilon(s) = \|\diamond \neg\Phi\|_\varepsilon(s)$$

$$\neg\|\diamond \Phi\|_\varepsilon(s) = \|\square \neg\Phi\|_\varepsilon(s)$$

Proof.

$$\begin{aligned} &\neg\|\square \Phi\|_\varepsilon(s) \\ &= \neg(\wedge_{t \in S, \alpha \in Act} (R(s, \alpha, t) \rightarrow \|\Phi\|_\varepsilon(t))) \\ &= \vee \neg(R(s, \alpha, t) \rightarrow \|\Phi\|_\varepsilon(t)) \\ &= \vee (\neg \neg R(s, \alpha, t) \wedge \neg \|\Phi\|_\varepsilon(t)) \\ &= \vee (R(s, \alpha, t) \wedge \|\neg\Phi\|_\varepsilon(t)) \\ &= \|\diamond \neg\Phi\|_\varepsilon(s) \end{aligned}$$

□

Similarly, Equation (2) can be proved.

Theorem 2. Given GPO_μ formula Φ , Φ_1 , and Φ_2 , functions $f_1, f_2: j^S \rightarrow j^S$ are defined as follows:

$$f_1(X)(s) = \|\Phi_2\|_\varepsilon(s) \vee \left(\|\Phi_1\|_\varepsilon(s) \wedge \bigvee_{s' \in S, \alpha \in Act} (R(s, \alpha, s') \wedge X(s')) \right),$$

$$f_2(X)(s) = \|\Phi\|_\varepsilon(s) \wedge \bigvee_{s' \in S, \alpha \in Act} (R(s, \alpha, s') \wedge X(s')).$$

where $X \in j^S, s \in S$. If $X, Y \in j^S$ and $X \subseteq Y$, then $f_1(X) \subseteq f_1(Y), f_2(X) \subseteq f_2(Y)$.

Proof. For a linear lattice $[0,1]$, the \wedge, \vee operations are monotonic. If $X \subseteq Y$, then for all $s \in S$, satisfy:

$$\begin{aligned} f_1(X)(s) &= \|\Phi_2\|_\varepsilon(s) \vee \left(\|\Phi_1\|_\varepsilon(s) \wedge \bigvee_{s' \in S, \alpha \in Act} (R(s, \alpha, s') \wedge X(s')) \right), \\ &\leq \|\Phi_2\|_\varepsilon(s) \vee \left(\|\Phi_1\|_\varepsilon(s) \wedge \bigvee_{s' \in S, \alpha \in Act} (R(s, \alpha, s') \wedge Y(s')) \right) = f_1(Y)(s). \end{aligned}$$

Hence, f_1 is monotonic, and f_2 is also monotonic, since f_1, f_2 is monotonic, and the possibility value $(0,1)$ is a finite complete lattice. According to Knaster-Tarski's theorem [11], f_1, f_2 have a least fixpoint and greatest fixpoint, respectively. Similiar to the classical μ -calculus, the CTL formulae for the generalized possibilistic decision process can also be expressed by the GPO_μ formulae. \square

Therefore, we obtain Theorem 3.

Theorem 3. The fixpoint semantics of CTL formulae for the generalized possibilistic decision process are expressed by GPO_μ formulae, as shown in the following:

$$\|E(\Phi_1 \cup \Phi_2)\| = \mu X. (\|\Phi_2\| \vee (\|\Phi_1\| \wedge \|\diamond X\|)) \tag{11}$$

$$\|EG\Phi\| = \|\nu X. \Phi \wedge \diamond X\|. \tag{12}$$

Take Equation (12) as an example, we prove the correctness of Theorem 3.

Proof. According to Theorem 2, the fixpoint semantics of CTL formulae for generalized possibilistic decision process are expressed by GPO_μ formulae, as shown in the following:

$$\|E(\Phi_1 \cup \Phi_2)\| = \mu X. (\|\Phi_2\| \vee (\|\Phi_1\| \wedge \|\diamond X\|))$$

$$\|EG\Phi\| = \|\nu X. \Phi \wedge \diamond X\|$$

\square

Take Equation (12) as an example; we prove the correctness of Theorem 3.

Proof. According to Theorem 2, $f_2(X)(s) = \Phi_\varepsilon(s) \wedge \bigvee_{s' \in S, \alpha \in Act} (R(s, \alpha, s') \wedge X(s')) = \nu X. (\|\Phi\| \wedge \|\diamond X\|)$ is monotonic. Let $A = \|EG\Phi\|, (A)(s) = \bigvee_{\pi \in \Pi(s)} \bigwedge_{i \in N} (\|\Phi\|(\pi(i)) \wedge R(\pi(i), \pi(i+1)))$ [24],

$$\begin{aligned} f_2(A)(s) &= \|\Phi\|(s) \wedge \bigvee_{t \in S} (R(s, \alpha, s_1) \wedge A(s_1)) \\ &= \|\Phi\|(s) \wedge \bigvee_{s_1 \in S} (R(s, \alpha, s_1) \wedge \bigvee_{\pi \in \Pi(s_1)} \bigwedge_{i \in N} (\|\Phi\|(\pi(i)) \wedge R(\pi(i), \alpha, \pi(i+1)))) \\ &= (\|\Phi\|(s) \bigvee_{s_1 \in S} (R(s, \alpha, s_1))) \bigvee_{\pi \in \Pi(s_1)} \bigwedge_{i \in N} (\|\Phi\|(\pi(i)) \wedge R(\pi(i), \alpha, \pi(i+1))) \end{aligned}$$

$$= \bigvee_{\pi \in \Pi(s)} \bigwedge_{i \in \mathbb{N}} (\|\Phi\|(\pi(i) \wedge R(\pi(i), \pi(i+1))))$$

$$= (A)(s)$$

$f_2(A)(s) = (A)(s)$, so the A is a fixpoint of f_2 .

Next, we prove that A is the greatest fixpoint of f_2 . Assume that Z is the fixpoint of

$$f_2, Z = \|\Phi\| \wedge \|\diamond Z\| = \|\Phi\| \wedge R \circ P_Z.$$

$$Z(s) = \|\Phi\|(s) \wedge \bigvee_{s_1 \in S} R(s, \alpha, s_1) \wedge Z(s_1)$$

$$\leq \|\Phi\|(s) \wedge \bigvee_{s_1, s_2 \in S} R(s, \alpha, s_1) \wedge \|\Phi\|(s_1) \wedge Z(s_1) \wedge R(s_1, \alpha, s_2) \wedge Z(s_2) \wedge \dots$$

$$\leq \bigvee_{\pi \in \Pi(s)} \bigwedge_{i \in \mathbb{N}} (\|\Phi\|(\pi(i) \wedge R(\pi(i), \pi(i+1))))$$

$$\leq (A)(s)$$

Hence,

$$A = \|\text{EG } \Phi\| \text{ is the greatest fixpoint of the function } f_2(X) = \|\Phi\| \wedge \|\diamond X\|.$$

□

4.2. Model-Checking Algorithm

Given a GPDP M , a state s , and a GPO_μ formula Φ , the purpose of GPO_μ model checking is to calculate the value of $\|\Phi\|_\varepsilon(s)$. For GPO_μ formula Φ , the value of $\|\Phi\|_\varepsilon(s)$ can be calculated recursively in $|\Phi|$ steps; $|\Phi|$ represents the length of the formula Φ [25], which are given as follows:

$$|p| = 1 |X| = 1$$

$$|j| = 1 | \neg \Phi | = |\Phi| + 1$$

$$|\Phi \vee \Psi| = |\Phi \wedge \Psi| = |\Phi| + |\Psi| + 1$$

$$|\diamond \Phi| = |\square \Phi| = |\Phi| + 1$$

$$|\mu X. \Phi| = |\nu X. \Phi| = |\Phi| + 2.$$

For formulas $\Phi = p, \Phi = X, \Phi = j, \Phi = \neg \Phi, \Phi = \Phi \vee \Psi, \Phi = \Phi \wedge \Psi$, the value of $\|\Phi\|_\varepsilon(s)$ can be calculated recursively according to Equations (1)–(6).

P_Φ denotes the column vector that the possibility that formula Φ holds on s , i.e., we have,

$$P_\Phi = (\|\Phi\|_\varepsilon(s))_{s \in S}. \tag{13}$$

To calculate the possibility about formula $\Phi = \diamond \Phi$, for any state and any $\alpha \in Act$, we convert the value of the formula into matrix operations,

$$\|\diamond \Phi\|_\varepsilon(s) = \bigvee_{t \in S, \alpha \in Act} (R(s, \alpha, t) \wedge \|\Phi\|_\varepsilon(t)) = R_\alpha \circ P_\Phi.$$

Therefore, the maximum and minimum values of $\|\diamond \Phi\|_\varepsilon(s)$ denoted by fuzzy matrices are shown below:

$$(\|\diamond \Phi\|_\varepsilon(s)_{max})_{s \in S} = R_{max} \circ P_\Phi \tag{14}$$

$$(\|\diamond \Phi\|_\varepsilon(s)_{min})_{s \in S} = R_{min} \circ P_\Phi. \tag{15}$$

For $\Phi = \Box\Phi$, according to Theorem 1, i.e., $\neg\|\Box\Phi\|_\varepsilon(s) = \|\Diamond\neg\Phi\|_\varepsilon(s)$, for any state and any $\alpha \in Act$, we have

$$\|\Box\Phi\|_\varepsilon(s) = \neg\|\Diamond\neg\Phi\|_\varepsilon(s) = \neg(R_\alpha \circ P_\Phi^c) = (R_\alpha \circ P_\Phi^c)^c \quad (16)$$

Thus, we use the fuzzy matrix to calculate the maximum and minimum values of $\|\Box\Phi\|_\varepsilon(s)$ as shown below:

$$(\|\Box\Phi\|_\varepsilon(s)_{max})_{s \in S} = (R_{min} \circ P_\Phi^c)^c \quad (17)$$

$$(\|\Box\Phi\|_\varepsilon(s)_{min})_{s \in S} = (R_{max} \circ P_\Phi^c)^c. \quad (18)$$

For $\Phi = \mu X.\Phi(X)$ and $\Phi = \nu X.\Phi(X)$, we use the fixpoint algorithm [21]; the corresponding algorithm is presented in Algorithm 1.

Algorithm 1: Fixpoint algorithm

Input: A function f from the possibility distributions on the state set S into itself.

Output: The fixpoint of f .

1. **Function** Fixpoint(Q, f)
 2. $Q' = f(Q)$
 3. **while** $Q \neq Q'$ **do**
 4. $Q = Q'$
 5. $Q' = f(Q)$
 6. **end while**
 7. **return** Q
 8. **End Function**
-

Furthermore, we respectively set $Q = 0$ or $Q = 1$ to calculate the least fixpoint or the greatest fixpoint in the initial recursion, which is different with the classical fixpoint algorithm.

Hence, according to the GPo_μ semantic, we give the GPo_μ model-checking algorithm, it is presented in Algorithm 2:

Algorithm 2: GPo_μ model checking algorithm

Input: A GPDP M and a GPo_μ formula Φ .

Output: For each state in M , the possibility $s \models \Phi$, i.e., $\|\Phi\|_\varepsilon(s)$.

1. **Function** Check (Φ)
 2. **Case** Φ
 3. p **return** $L(s, p)$
 4. X **return** $\varepsilon(X)$
 5. j **return** j
 6. $\neg\Phi$ **return** $1 - \|\Phi\|_\varepsilon(s)$
 7. $\Phi \vee \Psi$ **return** $\|\Phi\|_\varepsilon(s) \vee \|\Psi\|_\varepsilon(s)$
 8. $\Phi \wedge \Psi$ **return** $\|\Phi\|_\varepsilon(s) \wedge \|\Psi\|_\varepsilon(s)$
 9. $\Diamond\Phi$ **return** $R_\alpha \circ P_\Phi$
 10. $\Box\Phi$ **return** $(R_\alpha \circ P_\Phi^c)^c$
 11. $\mu X.\Phi(X)$ **return** Fixpoint ($0, f_\Phi$)
 12. $\nu X.\Phi(X)$ **return** Fixpoint ($1, f_\Phi$)
 13. **End Case**
 14. **End Function**
-

In the GPo_{μ} model-checking algorithm, for the formulas $\Phi = p$, $\Phi = X$, $\Phi = j$, $\Phi = \neg\Phi$, $\Phi = \Phi \vee \Psi$, $\Phi = \Phi \wedge \Psi$, the time complexity of the formulae is related to the length of Φ and the size of the system model M . For the formulae $\Phi = \diamond\Phi$, $\Phi = \square\Phi$, the time complexity of each formula is related to fuzzy matrix synthesis operation, i.e., $O(|S|^2)$. For fixpoint formulas $\mu X.\Phi(X)$ and $\nu X.\Phi(X)$, each fixpoint needs $O(n)$ iterations, $n = |S|$, the time for each iteration through the fuzzy matrix synthesis operation is $O(|S|^2)$, so the time complexity of the fixpoint formulae is $O(|S|^3)$.

5. An Illustrative Example and Case Study

5.1. An Illustrative Example

In this section, let us give an example used in [24] to illustrate the GPo_{μ} model-checking algorithm. Assume that a new type of disease occurs, and the doctors do not have enough knowledge to treat such diseases, they can only make treatment plans based on their own experience. Depending on the treatment options taken by the doctors, the patient’s physical health is also uncertain. So, GPDP is used to model the patient’s treatment process.

Assume that the doctors divide the patient’s status into three states, which are represented by s_0 , s_1 , and s_2 , respectively, that is, $S = \{s_0, s_1, s_2\}$, and the initial state of the patient is s_0 . $AP = \{P, G, E\}$ is the set of atomic propositions, where P, G, E respectively indicate that the patient’s health status is “poor”, “general”, and “excellent” in a certain state. For these three health conditions of patients, different doctors understand differently. Therefore, we give them a fuzzy value to indicate the degree of physical health. For example, $L(s_0, P) = 0.7$ denotes that the degree that the patient’s health is “poor” on state s_0 is 0.7; let $Act = \{\alpha, \beta, \gamma\}$, which means that the doctors treat the patient with α, β, γ : three different treatment schemes. $R(s_0, \beta, s_2) = 0.5$ denotes that the possibility that this patient’s health status changes from the state s_0 to s_2 is 0.5 after the doctors adopt the treatment scheme β .

From Figure 2, we can get the corresponding truth matrices of $P, G, E, P_P = \begin{pmatrix} 0.7 & 0.3 & 0.1 \end{pmatrix}^T$, $P_G = \begin{pmatrix} 0.15 & 0.24 & 0.65 \end{pmatrix}^T$, and $P_E = \begin{pmatrix} 0.2 & 0.51 & 0.8 \end{pmatrix}^T$. $R_{\alpha}, R_{\beta},$ and R_{γ} respectively denote the fuzzy matrices of three treatment schemes. The maximum possibilistic transition matrix is R_{max} , and the minimum possibilistic transition matrix is R_{min} .

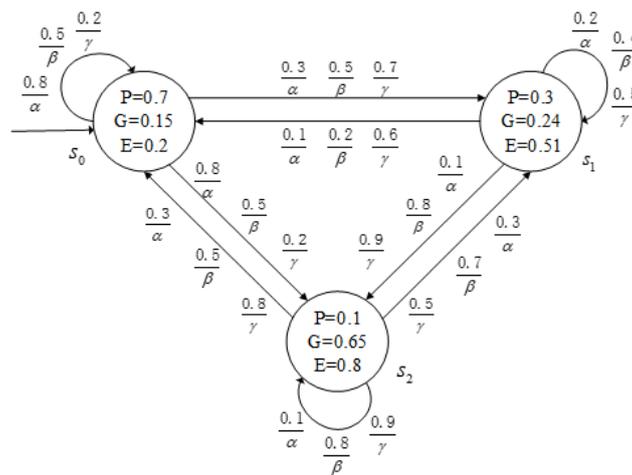


Figure 2. The GPDP of the patient treatment process.

$$R_{\alpha} = \begin{pmatrix} 0.8 & 0.3 & 0.8 \\ 0.1 & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.1 \end{pmatrix} \quad R_{\beta} = \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.8 \\ 0.5 & 0.7 & 0.8 \end{pmatrix} \quad R_{\gamma} = \begin{pmatrix} 0.2 & 0.7 & 0.2 \\ 0.6 & 0.5 & 0.9 \\ 0.8 & 0.5 & 0.9 \end{pmatrix}$$

$$R_{max} = \begin{pmatrix} 0.8 & 0.7 & 0.8 \\ 0.6 & 0.5 & 0.9 \\ 0.8 & 0.7 & 0.9 \end{pmatrix} \quad R_{min} = \begin{pmatrix} 0.2 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.1 \end{pmatrix}$$

Some calculations are presented as follows in detail.

$\| \diamond E \|_{\varepsilon}(s)$ denotes the possibility that the patient’s health eventually changes to “excellent” after a treatment.

$\| \diamond E \|_{\varepsilon}(s)_{max}$, $\| \diamond E \|_{\varepsilon}(s)_{min}$ respectively represent the maximum and minimum possibilities of $\| \diamond E \|_{\varepsilon}(s)$ in three states.

$$\| \diamond E \|_{\varepsilon}(s)_{max} = R_{max} \circ P_E = \begin{pmatrix} 0.8 & 0.8 & 0.8 \end{pmatrix}^T,$$

$$\| \diamond E \|_{\varepsilon}(s)_{min} = R_{min} \circ P_E = \begin{pmatrix} 0.3 & 0.2 & 0.3 \end{pmatrix}^T.$$

Hence,

$$\| \diamond E \|_{\varepsilon}(s_0)_{max} = 0.8, \| \diamond E \|_{\varepsilon}(s_1)_{max} = 0.8, \| \diamond E \|_{\varepsilon}(s_2)_{max} = 0.8.$$

$$\| \diamond E \|_{\varepsilon}(s_0)_{min} = 0.3, \| \diamond E \|_{\varepsilon}(s_1)_{min} = 0.2, \| \diamond E \|_{\varepsilon}(s_2)_{min} = 0.2.$$

$\| \square E \|_{\varepsilon}(s)$ denotes the possibility that the patient’s health always changes to “excellent” after a treatment.

$\| \square E \|_{\varepsilon}(s)_{max}$ and $\| \square E \|_{\varepsilon}(s)_{min}$ respectively represent the maximum and minimum possibilities of $\| \square E \|_{\varepsilon}(s)$ in three states.

$$\| \square E \|_{\varepsilon}(s)_{max} = (R_{min} \circ P_E^c)^c = \begin{pmatrix} 0.7 & 0.8 & 0.7 \end{pmatrix}^T,$$

$$\| \square E \|_{\varepsilon}(s)_{min} = (R_{max} \circ P_E^c)^c = \begin{pmatrix} 0.2 & 0.4 & 0.2 \end{pmatrix}^T.$$

Hence,

$$\| \square E \|_{\varepsilon}(s_0)_{max} = 0.7, \| \square E \|_{\varepsilon}(s_1)_{max} = 0.8, \| \square E \|_{\varepsilon}(s_2)_{max} = 0.7.$$

$$\| \square E \|_{\varepsilon}(s_0)_{min} = 0.2, \| \square E \|_{\varepsilon}(s_1)_{min} = 0.4, \| \square E \|_{\varepsilon}(s_2)_{min} = 0.2.$$

According to Formula (11), $\| E(\Phi_1 \cup \Phi_2) \| (s)$ is the least fixpoint of $\tau(X) = \Phi_2 \vee (\Phi_1 \wedge \diamond X)$. Let $P_X = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$, $\| E(P \cup E) \| (s)$ means the possibility that the patient’s health eventually changes from “poor” to “excellent”.

The process of calculating the maximum and minimum possibilities with fuzzy matrices is as follows:

$$\tau(X_1) = P_E \vee (P_p \wedge (R_{max} \circ P_X)) = \begin{pmatrix} 0.2 & 0.51 & 0.8 \end{pmatrix}^T,$$

$$\tau(X_2) = P_E \vee (P_p \wedge (R_{max} \circ P_{X_1})) = \begin{pmatrix} 0.7 & 0.51 & 0.8 \end{pmatrix}^T,$$

$$\tau(X_3) = P_E \vee (P_p \wedge (R_{max} \circ P_{X_2})) = \begin{pmatrix} 0.7 & 0.51 & 0.8 \end{pmatrix}^T,$$

$$\tau(X_2) = \tau(X_3), \| E(P \cup E) \| (s)_{max} = \begin{pmatrix} 0.7 & 0.51 & 0.8 \end{pmatrix}^T.$$

Similarly,

$$\| E(P \cup E) \| (s)_{min} = \begin{pmatrix} 0.3 & 0.51 & 0.8 \end{pmatrix}^T.$$

Hence,

$$\| E(P \cup E) \| (s_0)_{max} = 0.7, \| E(P \cup E) \| (s_1)_{max} = 0.51, \| E(P \cup E) \| (s_2)_{max} = 0.8.$$

$$\| E(P \cup E) \| (s_0)_{min} = 0.3, \| E(P \cup E) \| (s_1)_{min} = 0.51, \| E(P \cup E) \| (s_2)_{min} = 0.8.$$

According to Formula (12), $\| EG \Phi \| (s)$ is the greatest fixpoint of $\tau(X) = \Phi \wedge \diamond X$. Let $P_X = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$, $\| EG E \| (s)$ means the possibility that the patient’s health is potentially always “excellent”.

The process of calculating the maximum and minimum possibilities by fuzzy matrices is as follows:

$$\begin{aligned} \tau(X_1) &= P_E \wedge (R_{max} \circ P_X) = \begin{pmatrix} 0.2 & 0.51 & 0.8 \end{pmatrix}^T, \\ \tau(X_2) &= P_E \wedge (R_{max} \circ P_{X_1}) = \begin{pmatrix} 0.2 & 0.51 & 0.8 \end{pmatrix}^T, \\ \tau(X_1) &= \tau(X_2), \|EG E\|(s)_{max} = \begin{pmatrix} 0.2 & 0.51 & 0.8 \end{pmatrix}^T. \end{aligned}$$

Similarly,

$$\|EG E\|(s)_{min} = \begin{pmatrix} 0.2 & 0.2 & 0.2 \end{pmatrix}^T.$$

Hence,

$$\begin{aligned} \|EG E\|(s_0)_{max} &= 0.2, \|EG E\|(s_1)_{max} = 0.51, \|EG E\|(s_2)_{max} = 0.8, \\ \|EG E\|(s_0)_{min} &= 0.2, \|EG E\|(s_1)_{min} = 0.2, \|EG E\|(s_2)_{min} = 0.2. \end{aligned}$$

As a result, the possibility values that each state satisfies the attributes are obtained. Through these data, doctors can compare the results of different treatment options and then make the practical treatment scheme. Our algorithm provides data support for doctors' decision analysis.

5.2. Case Study

We use the intelligent washing machine studied in [18] as an example to better explain the application of the GPo_μ model-checking method. As shown in Figure 3, it is the control system of the intelligent washing machine, where S₀ to S₁₀ are the states of the system, there is only one action in the system, and the possibility values are marked between states. The atomic propositions are dirty (*di*), detergent (*de*), and running (*r*), which denote the condition of the cloths, the state of the detergent, and the state of the system. The value of every atomic proposition is labeled at each state. For example, *di* = 1 means that the clothes are very dirty, *di* = 0.5 means that the clothes are moderately dirty, and *di* = 0 means that the clothes are clean. *de* = 1 means there is a lot of detergent, *de* = 0.5 means there is moderate detergent, and *de* = 0 means that there is no detergent. *r* = 1 denotes that the machine is running, and *r* = 0 means that the washing machine is off. The initial state is S₀, and the clothes are very dirty, there is a lot of detergent, and the system is off. From S₁ to S₂, it can be found that the dosage of the detergent is decreasing. At S₃, the detergent is 0, but the clothes are still dirty, and the detergent should be added, so S₃ may return to initial state S₀ or return to S₂. In addition, each state has a self-circulation, and the possibility is 1, which are omitted in the figure.

For the intelligent washing machine system, we convert some properties into GPo_μ formulae:

- Property1** The possibility that the washing machine can be turned off at the next time, which is denoted as $\diamond \neg \textit{running}$.
- Property2** The possibility that the clothes can be cleaned when the washing machine is in the next state, which is denoted as $\square \neg \textit{dirty}$.
- Property3** The possibility that the washing machine may have detergent before the clothes are clean, which is denoted as $\mu X. \neg \textit{dirty} \vee (\textit{deterget} \wedge \diamond X)$.
- Property4** The possibility that detergent is always in the washing machine, which can be denoted as $\nu X. \textit{deterget} \wedge \diamond X$.
- Property5** The possibility that only with detergent, the clothes can be cleaned in the washing machine, which is denoted as $\nu X. (\textit{deterget} \rightarrow \neg \textit{dirty}) \wedge \diamond X$.

The results of verifying the properties of the intelligent washing machine system you can see in the following Table 1.

Table 1. Results of verifying the properties of the intelligent washing machine system.

Properties	GPO _μ Fomulas	Verifying Results
Property1	$\diamond \neg \text{running}$	$(1,0,0,1,0,0,0,1,0,0)^T$
Property2	$\square \neg \text{dirty}$	$(0,0,0,0,0.5,0.5,0.5,0,1,1)^T$
Property3	$\ \mu X. \neg \text{dirty} \vee (\text{deterget} \wedge \diamond X)\ $	$(0.5,0.5,0.5,0,0.5,0.9,0.5,1,1,1)^T$
Property4	$\vee X. \text{deterget} \wedge \diamond X$	$(1,1,0.5,0,0.5,1,0,0,0.5,1)^T$
Property5	$\vee X. (\text{deterget} \rightarrow \neg \text{dirty}) \wedge \diamond X$	$(0,0,0.5,1,0.5,0.5,1,1,1,1)^T$

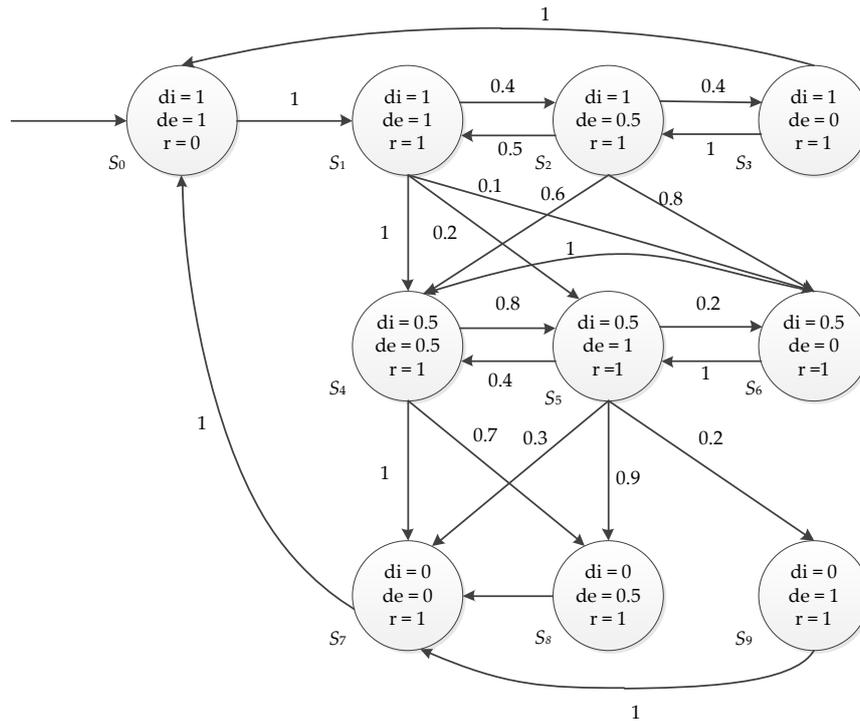


Figure 3. The GPDP of the intelligent washing machine system.

The table lists the possibility value that each state meets the properties. For example, for Property 1, $\|\diamond \neg \text{running}\|(S_0) = \|\diamond \neg \text{running}\|(S_3) = \|\diamond \neg \text{running}\|(S_7) = 1$, the result indicates that only at states S_0, S_3, S_7 can the washing machine be turned off at the next time. For Property 3, we know $\|\mu X. \neg \text{dirty} \vee (\text{deterget} \wedge \diamond X)\|(S_7) = 0$, which indicates the possibility that the washing machine may have detergent before the clothes only at S_7 are clean is 0, because the clothes are clean and there is no detergent, so S_7 does not need to satisfy Property 3. For Property 5, $\|\vee X. \text{deterget} \wedge \diamond X\|(S_0) = \|\vee X. \text{deterget} \wedge \diamond X\|(S_1) = \|\vee X. \text{deterget} \wedge \diamond X\|(S_5) = \|\vee X. \text{deterget} \wedge \diamond X\|(S_9) = 1$, which indicates that detergent is potentially always in the washing machine at states S_0, S_1, S_5 , and S_9 .

In [18], they use the GPoLTL formulae to represent the properties of the systems and verified the fuzzy linear-time properties of systems. Different from them, we denoted the properties of the systems by GPO_μ formulas, and we can express different properties of the systems that the GPoLTL formulae can not express.

6. Conclusions

This paper introduced the μ -calculus model-checking algorithm for the generalized possibilistic decision process, which is an extension of the L_μ model checking in [26]. We first give the generalized possibilistic decision process, and then extend the classical μ -calculus to describe the complex logical relationships and attribute the characteristics of nondeterministic systems. We simplify the model-checking problem into fuzzy matrix operations. It implements attribute verification for

nondeterministic systems, and we have explained some examples to prove the effectiveness and practicability of the algorithm.

The classical μ -calculus model-checking algorithm simply searches the finite state space of Boolean transition systems to find the satisfaction set of system properties. The GPo_μ model-checking algorithm is more powerful in terms of expressiveness for expressing possible information, because it can characterize uncertain information for dealing with the incompleteness of the information in the systems and specifications, and it can give the degree to which the systems satisfy the specifications. The satisfaction relation becomes the possibility values that expected properties hold, which is more in line with the characteristics of the actual systems.

In the actual system design, for a system containing possibility information, the system is uncertain (lack of basic information) or inconsistent (conflicts often occur when collecting information from multiple sources Information), and classical logic is unable to infer and calculate the relevant characteristics of uncertain systems. Our methods can verify the properties of nondeterministic systems, such as expert systems and intelligent control uncertain systems, etc. However, the research on the expression ability of extended μ -calculus is not enough, such as multilayer nesting between multiple fixpoint formulas, and research on the optimization of nesting algorithms is not enough. In the future, we will continue to study the expression ability of the algorithm and develop a practical model checker to implement automatic verification.

Author Contributions: Conceptualization, J.J.; methodology, J.J., P.Z., and Z.M.; software, P.Z. and Z.M.; validation, J.J., P.Z., and Z.M.; formal analysis, J.J., P.Z., and Z.M.; investigation, J.J.; resources, J.J., P.Z., and Z.M.; data curation, J.J., P.Z., and Z.M.; writing—original draft preparation, J.J. and Z.M.; writing—review and editing, J.J., P.Z., and Z.M.; visualization, P.Z.; supervision, J.J.; project administration, Z.M.; funding acquisition, J.J., P.Z., and Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of China (No. 61762002, No. 61962001) and the Ningxia Natural Science Foundation (No. 2018AAC03127).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baier, C.; Katoen, J.P. *Principles of Model Checking*; The MIT Press: Cambridge, MA, USA, 2008.
2. Clarke, E.M.; Grumberg, O.; Peled, D. *Model Checking*; The MIT Press: Cambridge, MA, USA, 1999.
3. Stephane, D.; Valentin, G.; Martin, L. *Temporal Logics in Computer Science*; Cambridge University Press: Cambridge, UK, 2016.
4. Clarke, E.M.; Henzinger, T.A.; Veith, H.; Bloem, R. (Eds.) *Handbook of Model Checking*; Springer: Cham, Switzerland, 2018.
5. de Alfaro, L. Quantitative Verification and Control via the Mu-Calculus. In *International Conference on Concurrency Theory*; Springer: Berlin/Heidelberg, Germany, 2003.
6. McIver, A.; Carroll, M. *Results on the quantitative μ -calculus $qM \mu$* ; ACM Transactions on Computational Logic (TOCL): New York, NY, USA, 2007; Available online: <https://dl.acm.org/citation.cfm?doid=1182613.1182616>.
7. Li, Y.; Li, Y.; Ma, Z. Computation tree logic model checking based on possibility measures. *Fuzzy Sets Syst.* **2015**, *262*, 44–59. [[CrossRef](#)]
8. Yuan, S.; Wei, J.L.; Li, Y. Model checking of generalized possibilistic computation tree logic with multi-valued decision process. *Comput. Eng. Sci.* **2019**, *41*, 88–97.
9. Lei, L.; Guo, Y.; Zhang, Y.B. Computation tree logic symbolic model checking under possibility measures. *Comput. Eng. Sci.* **2018**, *40*, 106–112.
10. Fischer, D.; Gradel, E.; Kaiser, L. Model checking games for the quantitative mu-calculus. *Theory Comput. Syst.* **2010**, *47*, 696–719. [[CrossRef](#)]
11. Li, Y.; Droste, M.; Lei, L. Model checking of linear-time properties in multi-valued systems. *Inf. Sci.* **2017**, *377*, 51–74. [[CrossRef](#)]
12. Fainekos, G.E. *An Introduction to Multi-Valued Model Checking: MS-CIS-05-16*; University of Pennsylvania: Philadelphia, PA, USA, 2005.

13. Narasimha, M.; Cleaveland, R.; Iyer, P. Probabilistic temporal logics via the modal μ -calculus. *Theor. Comput. Sci.* **1998**, *342*, 316–350.
14. Chechik, M.; Devereux, B.; Easterbrook, S.; Gurfinkel, A. Multi-valued symbolic model checking. *ACM Trans. Softw. Eng. Methodol.* **2003**, *12*, 371–408. [[CrossRef](#)]
15. Gurfinkel, A.; Chechik, M. Multi-Valued Model Checking via Classical Model Checking. In *CONCUR 2003—Concurrency Theory, Proceedings of the 14th International Conference on Concurrency Theory, Marseille, France, 3–5 September 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 263–277.
16. Mallya, A. Deductive multi-valued model checking. In *Proceedings of the International Conference on Logic Programming, Sitges, Spain, 2–5 October 2005*.
17. Pan, H.; Li, Y.; Cao, Y.; Ma, Z. Model checking computation tree logic over finite lattices. *Theor. Comput. Sci.* **2016**, *612*, 45–62. [[CrossRef](#)]
18. Li, Y. Quantitative model checking of linear-time properties based on generalized possibility measures. *Fuzzy Sets Syst.* **2017**, *320*, 17–39. [[CrossRef](#)]
19. Li, Y.; Lei, L.; Li, S. Computation tree logic model checking based on multi-valued possibility measures. *Inf. Sci.* **2019**, *485*, 87–113. [[CrossRef](#)]
20. Li, Y.; Li, L. Model checking of linear-time properties based on possibility measure. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 842–854. [[CrossRef](#)]
21. Li, Y. Quantitative computation tree logic model checking based on generalized possibility measures. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 2034–2047. [[CrossRef](#)]
22. Klein, J.; Baier, C.; Chrszon, P.; Daum, M.; Dubstlatt, C.; Klüppelholz, S.; Märcker, S.; Müller, D. Advances in Symbolic Probabilistic Model Checking with PRISM. In *Proceedings of the 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Eindhoven, The Netherlands, 2–8 April 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 349–366.
23. Li, Y.; Li, P. *Theory of Fuzzy Computation*; Science Press: Beijing, China, 2016.
24. Ma, Z.; Li, Y. Model checking of generalized possibilistic computing tree logic based on decision process. *Chin. Sci. Inf. Sci.* **2016**, *46*, 1591–1607.
25. Jungteerapanich, N. A Tableau System for the Modal μ -Calculus. In *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX), Oslo, Norway, 6–10 July 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 220–234.
26. Zhang, P.; Jiang, J.; Ma, Z.; Zhu, H. Quantitative μ -calculus model checking algorithm based on generalized possibility measures. In *Proceedings of the 2019 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress, Fukoka, Japan, 5–8 August 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 449–453.

