

Model Checking Fuzzy Computation Tree Logic Based on Fuzzy Decision Processes with Cost

Zhanyou Ma *, Zhaokai Li *, Weijun Li, Yingnan Gao and Xia Li

School of Computer Science and Engineering, North Minzu University, Yinchuan 750000, China

* Correspondence: mazhany@nmu.edu.cn (M.Z.); zhaokli@126.com (L.Z.)

Abstract: In order to solve the problems in fuzzy computation tree logic model checking with cost operator, we propose a fuzzy decision process computation tree logic model checking method with cost. Firstly, we introduce a fuzzy decision process model with cost, which can not only describe the uncertain choice and transition possibility of systems, but also quantitatively describe the cost of the systems. Secondly, under the model of the fuzzy decision process with cost, we give the syntax and semantics of the fuzzy computation tree logic with cost operators. Thirdly, we study the problem of computation tree logic model checking for fuzzy decision process with cost, and give its matrix calculation method and algorithm. We use the example of medical expert systems to illustrate the method and model checking algorithm.

Keywords: fuzzy model checking; fuzzy decision processes; fuzzy computation tree logic; cost operator



Citation: Ma, Z.; Li, Z.; Li, W.; Gao, Y.; Li, X. Model Checking Fuzzy Computation Tree Logic Based on Fuzzy Decision Processes with Cost. *Entropy* **2022**, *24*, 1183. <https://doi.org/10.3390/e24091183>

Academic Editor: Éloi Bossé

Received: 10 August 2022

Accepted: 23 August 2022

Published: 24 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Model checking is an important formal verification method. Because of its automatic, model checking has been widely used in the analysis and verification of computer hardware and software systems, communication protocols, security protocols and so on. Model checking is mainly composed of three parts: the first is to model the system under consideration, the second is to use formal language to describe the properties, and the third is to use a model checking algorithm to systematically check whether or not the given model satisfies these properties [1].

Classical model checking [2,3] was formulated for verifying the qualitative properties of systems. However, the Boolean result is not enough for the models with quantitative information, such as a 90 percent probability of the system crashing during operation. At present, more and more complex computer systems have the characteristics of randomness, uncertainty and inconsistency. In order to deal with the verification of complex systems, many quantitative model checking methods have been proposed by academia.

Probabilistic model checking [4–8] mainly deals with the problem of model checking for systems with uncertainties generated by stochastic processes. Its goal is to determine the accuracy of probabilistic systems for quantitative probability specifications. Sometimes models may contain inconsistencies as they connect conflict points or contain components designed by different designers independently. In order to verify complex systems with inconsistencies and uncertainties, multi-valued model checking [9–12] is proposed. Fuzzy model checking [13–20] pays more attention to the true value of the properties, which is another kind of uncertainty, caused by unclear concept extension [21–23]. Both possibility model checking [13,14], and generalized possibility model checking [15,16,24] are based on possibility measure, a combination of possibility measure theory in fuzzy set with model checking. Possibilistic Kripke structure is used to model the system, and possibilistic temporal logic is used to describe properties. Li Yongming et al. [14] use the operators in possibilistic computation tree logic to replace the existence and arbitrary quantifiers

of classical computation tree logic to calculate the possibility that the model satisfies the properties. In the process of calculating the possibility measure, the possibility of the path cylinder set after reaching the state also participates in the calculation, but these calculations can be ignored in most systems. Pan Haiyu et al. [19] use fuzzy Kripke structure to model the systems, fuzzy computation tree logic (FCTL) to describe properties and study fuzzy model checking.

Fuzzy Kripke structure is characterized by a state-to-state transition without a cost. However, in daily life, the transition may be different and have some cost [25]. For example, consider the disease diagnosis system studied in [14,19,26]. Suppose there are multi-steps treatments A and B for a disease, and different treatment needs different costs for each step during the process. The models in the literature [14,19,26] can only describe the situation that experts have been using treatment A or B during the treatment but cannot describe the situation where experts use A as the first and third steps and use B as the other steps. In addition, the cost of treatments is also unable to represent and verify. For the above reasons, we have done this paper. First, we define a fuzzy decision process model with a cost function, which can not only describe the nondeterministic choices but also describe the quantitative properties. Second, we introduce the definition of scheduler into the uncertain selection of actions so that the fuzzy decision process with cost function can be transformed into a fuzzy Kripke structure with a cost function. Then, we present the syntax and semantics of fuzzy computation tree logic with the cost operator. Finally, we calculate the quantitative possibility and cost of the problem according to the model checking algorithm. The main contributions of this paper are as follows.

- A fuzzy decision process model with cost function is defined, which can describe the cost and other quantitative properties of a fuzzy system. The action property in the model is used to describe the uncertain action selection of the model, and the cost property is used to describe the cost of the system.
- The FCTL is extended to FCTL with cost operator. The fuzzy computation tree logic with cost operator inherits the existence and arbitrary quantifiers of classical temporal logic and adds operators about cost.
- The fuzzy computation tree logic model checking quantitative calculation formula and algorithm are given. At the same time, the complexity of the algorithm is analyzed.

The paper is organized as follows: in Section 2, the basic theoretical knowledge of fuzzy mathematics and fuzzy Kripke structure are given. In Section 3, we define a fuzzy decision process model with a cost function. In Section 4, we define fuzzy computation tree logic with a cost operator. In Section 5, we give the fuzzy computation tree logic model checking quantitative calculation formula and algorithm. Section 6 is an example. Section 7 summarizes this paper.

2. Preliminaries

A fuzzy set is a mathematical concept proposed by Zadeh in 1965. Fuzziness, in general, refers to any indistinct phenomena, where there is no clear boundary between “stability” and “instability”, “healthy” and “unhealthy”. The transition from one state to another is a continuous process when quantitative changes accumulate and eventually result in a qualitative change, which is due to the uncertainty caused by the breaking of the law of excluded middle. To model and verify fuzzy systems, we provide some necessary knowledge, which includes the fuzzy set, fuzzy set operation, fuzzy matrix operation, closure and others.

Definition 1 ([27]). *Let X be a universal set. A fuzzy set A of X is a function which associates each element in X a value in the interval $[0, 1]$, i.e., $A : X \rightarrow [0, 1]$. For $x \in X$, $A(x)$ is the membership of x in the fuzzy set A .*

We use $\mathcal{F}(X)$ to represent all fuzzy sets in X , i.e., $\mathcal{F}(X) = \{A \mid A : X \rightarrow [0, 1]\}$.

Definition 2 ([27]). Let $A, B \in \mathcal{F}(X)$, we use $A \cup B, A \cap B, A^c$ to represent the union, intersection and complement of A and B . The definition is as follows.

$$\begin{aligned} (A \cup B)(x) &= A(x) \vee B(x) = \max\{A(x), B(x)\}, \\ (A \cap B)(x) &= A(x) \wedge B(x) = \min\{A(x), B(x)\}, \\ A^c(x) &= 1 - A(x). \end{aligned}$$

Furthermore, we have De Morgan’s laws.

$$(A \cup B)^c = A^c \cap B^c, (A \cap B)^c = A^c \cup B^c.$$

Fuzzy matrix is a kind of special matrix, in which the value of each element is in the interval $[0, 1]$. It has some interesting operations and natures as follows.

Definition 3 ([28]). Let R and S be two fuzzy matrixes with m rows and n columns, i.e., $R = (r_{ij})_{m \times n}, S = (s_{ij})_{m \times n}$.

The standard operations on fuzzy matrixes R, S are defined in the following manner:

$$\begin{aligned} R = S, & \text{ if and only if } r_{ij} = s_{ij} \text{ for all } i, j. \\ R \subseteq S, & \text{ if and only if } r_{ij} \leq s_{ij} \text{ for all } i, j. \\ R \cup S &= (r_{ij} \vee s_{ij})_{m \times n}. \\ R \cap S &= (r_{ij} \wedge s_{ij})_{m \times n}. \\ R^c &= (1 - r_{ij})_{m \times n}. \end{aligned}$$

Definition 4 ([28]). Let R be a fuzzy matrix with m rows and n columns, S be a fuzzy matrix with n rows and l columns, i.e., $R = (r_{ij})_{m \times n}, S = (s_{ij})_{n \times l}$. The composition operation of R and S is $R \circ S = (t_{ij})_{m \times l}$, where $t_{ij} = \bigvee_{k=1}^n (r_{ik} \wedge s_{kj})$, ($i = 1, 2, \dots, m, j = 1, 2, \dots, l$). For fuzzy matrixes R, S, T the composition operation has some laws.

$$(R \circ S) \circ T = R \circ (S \circ T);$$

$$(R \cup S) \circ T = (R \circ T) \cup (S \circ T).$$

Let X be a universal set. For the fuzzy matrix $R = (R(s, t))_{s, t \in X}$, we use R^+ to denote its transitive closure. When X is finite, and X has $|X|$ elements, then $R^+ = R \cup R^2 \cup \dots \cup R^{|X|}$ [29], where $R^{k+1} = R^k \circ R$ for any positive integer number k . The Kleene closure $R^* = R^0 \cup R^+$, for each $1 \leq s, t \leq |S|, R^0(s, t) = \begin{cases} 1 & s = t \\ 0 & s \neq t \end{cases}$.

Transition systems or Kripke structures are the key models for model checking. Corresponding to fuzzy model checking, we expand the notion of fuzzy Kripke structures, defined as follows.

Definition 5 ([19]). A fuzzy Kripke structure(FKS) is a tuple $\mathcal{K} = (S, P, I, AP, L)$, where

- S is a countable, non-empty set of states,
- $P : S \times S \rightarrow [0, 1]$ is the fuzzy transition. For each $s \in S$, there exist $t \in S$ such that $P(s, t) > 0$,
- $I : S \rightarrow [0, 1]$ is the initial fuzzy distribution function. The initial state is s , and the truth value is $I(s)$,
- AP is the set of atomic propositions,
- $L : S \times AP \rightarrow [0, 1]$ is a fuzzy labeling function. $L(s, p)$ is the truth value to the atomic proposition p in state s .

The transition of FKS is certain for a pair of states, i.e., $P(s, t)$ is unique. However, on many occasions, we can transmit from one state to another by many methods. In other words, $P(s, t)$ is not certain. We carry out the fuzzy decision processes with a cost which are uncertain in the transition and have a function of cost. For the conditions of daily life, we use the natural number set \mathbb{N} as the range of the cost function as an example.

3. Fuzzy Decision Processes with Cost

Fuzzy systems are often used to describe the medical expert systems. Due to the different judgment standards of each expert on the patient’s condition and treatment effect, it establishes the model better. At the same time, there are many new problems caused by a variety of treatment options for the same disease. For instance, how to choose the best treatment option in a variety of options? How to evaluate the cost of various treatment options? FKS cannot model the interleaving behavior and the cost of concurrent processes in an adequate manner. For this purpose, we extend FKS to an uncertain system model with cost. The specific definition is as follows.

Definition 6. A fuzzy decision process with cost (FDPC) is a tuple $M_f = (S, Act, P, I, AP, L, C)$, where

- S is a countable, non-empty set of states,
- Act is the set of actions,
- $P : S \times Act \times S \rightarrow [0, 1]$ is the fuzzy transition. For each $s \in S$ and $\alpha \in Act$, there exists $t \in S$ which let $P(s, \alpha, t) > 0$,
- $I : S \rightarrow [0, 1]$ is the initial fuzzy distribution function. For $s \in S$, the truth value is $I(s)$,
- AP is the set of atomic propositions,
- $L : S \times AP \rightarrow [0, 1]$ is a fuzzy labeling function. $L(s, p)$ is the truth value to the atomic proposition p in state s .
- $C : S \times Act \rightarrow \mathbb{N}$ is a cost function. For each $s \in S$ and $\alpha \in Act$, $C(s, \alpha)$ is the cost of that the action α is selected in state s .

If S , Act and AP are finite, we say the M_f is finite. We say that action α is enabled in state s if there exists a state $t \in S$ such that $P(s, \alpha, t) > 0$. $Act(s)$ denotes the set of actions which can be enabled in state s .

$\hat{\pi} = s_0\alpha_0s_1\alpha_1s_2\dots s_{n-1}\alpha_{n-1}s_n$ denotes a finite path of M_f , and $\pi = s_0\alpha_0s_1\alpha_1s_2\dots \in (S \times Act)^\omega$ denotes an infinite path of M_f . $Paths(s)$ denotes the set of the infinite paths which begin from state s . $Path^{fin}(M_f)$ denotes the set of finite paths which begin from all states of M_f . $Paths(M_f)$ is the set of infinite paths which begin from all initial states of M_f .

Example 1. Figures 1–4 is a simple, in which there are three transitions represented by α , β and γ . The model has three states s_0 , s_1 and s_2 . The variables in the state indicate the atomic proposition a and b . Different states have different memberships of atomic propositions. Therefore, we use fuzzy values to describe them. When action α_i is used in state s_i , cost function $C(s_i, \alpha_i)$ is generated. When using a single transition α or β or γ , the FKSs are shown in Figures 1–3. The FDPC produced by them is shown in Figure 4. The transition possibility is given by the number on the connecting line, and the cost is indicated by the underlined number in the figure. For example, $s_0 \xrightarrow{\alpha, 0.8, \underline{160}} s_1$ indicates that in state s_0 , using treatment scheme α , the possibility of transition to state s_1 is 0.8, and the treatment cost is 160.

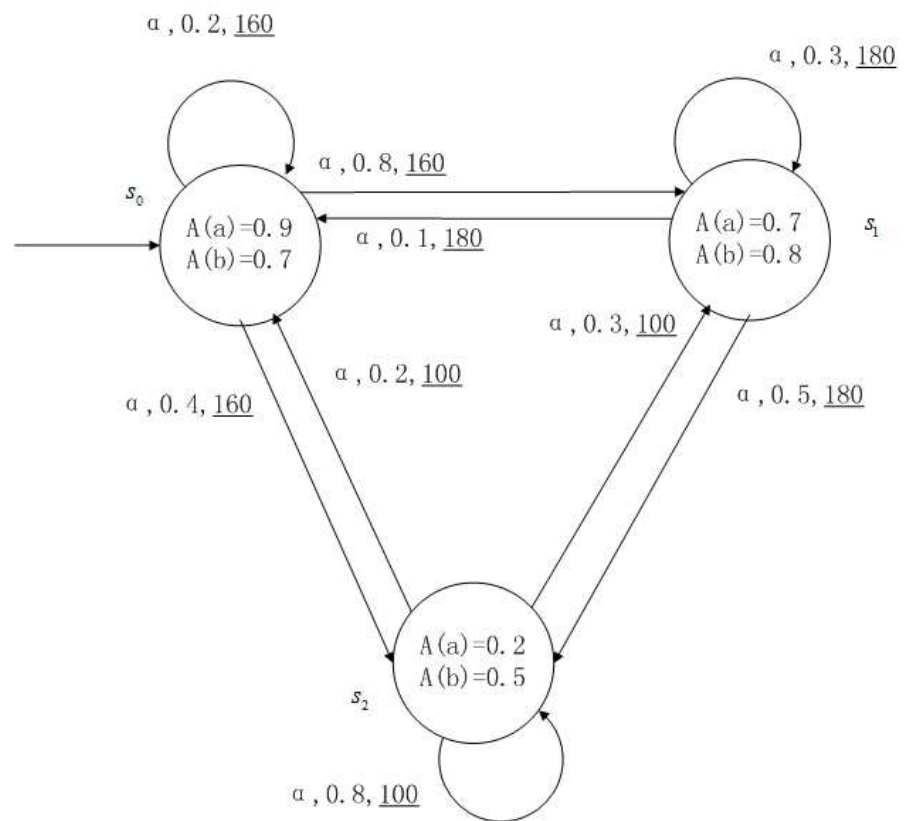


Figure 1. FKS transmitted by α .

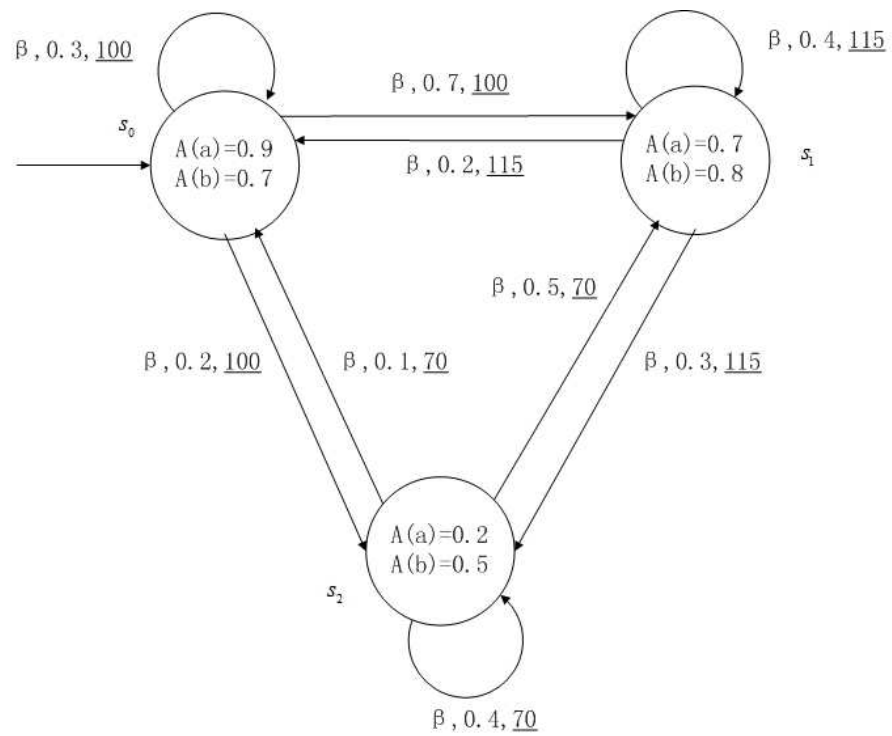


Figure 2. FKS transmitted by β .

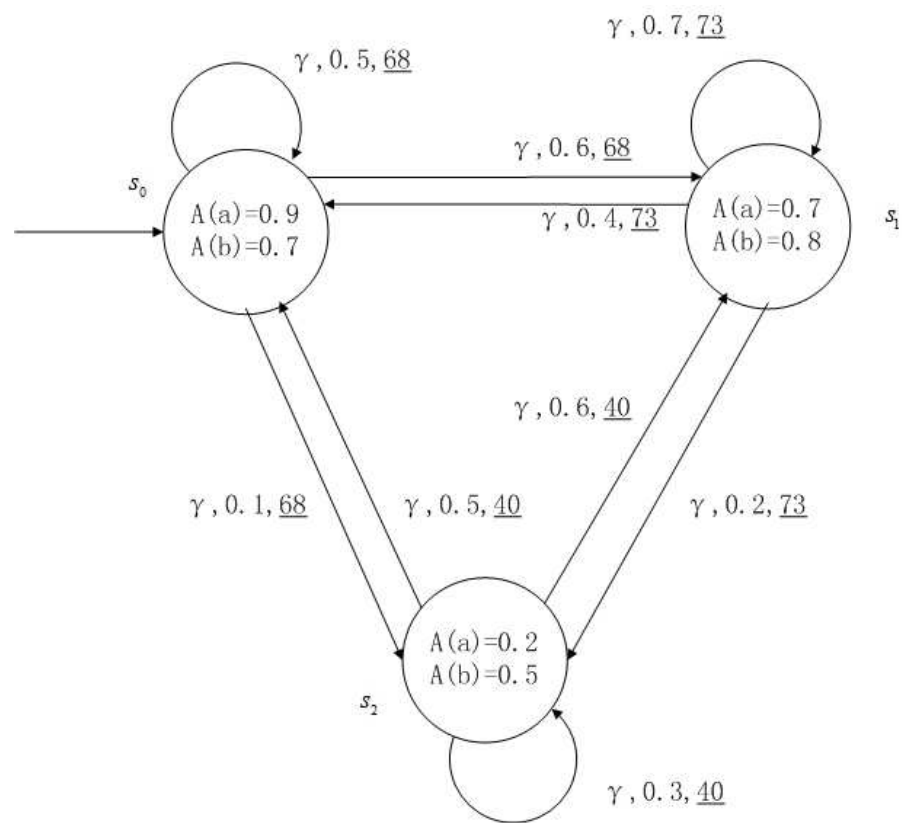


Figure 3. FKS transmitted by γ .

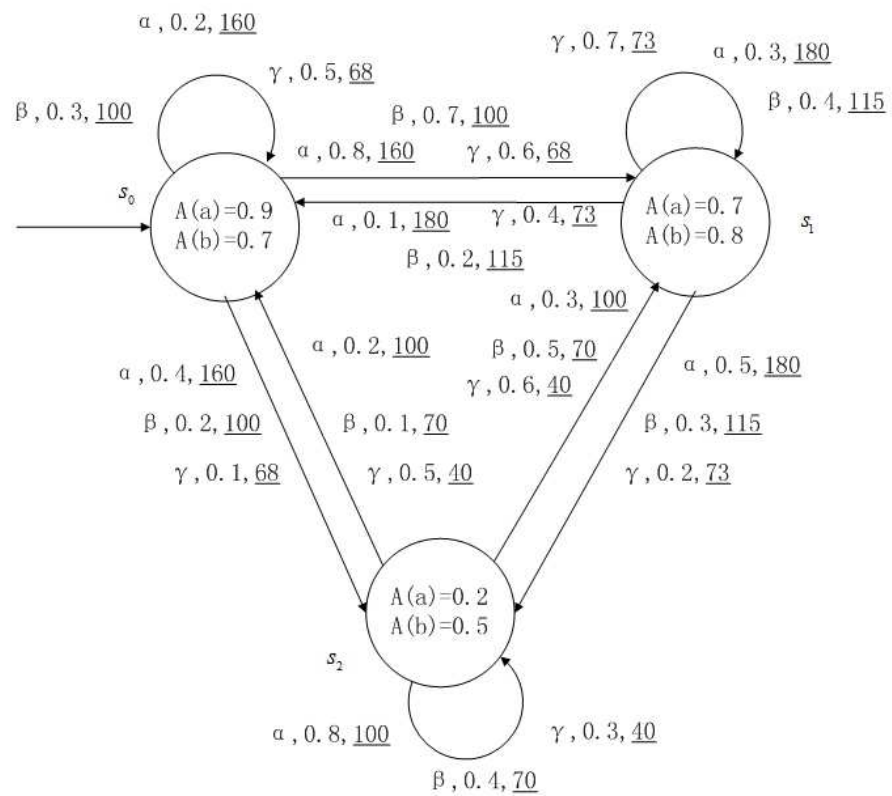


Figure 4. FDPC created by above FKSs.

FDPCs are more complex than FKSs because of the interleaving of the transitions. For example, for a states sequence $s_0s_1s_2$, there is only one possibility in an FKS, but it may be multiple possibilities in a FDPC, such as $s_0\alpha s_1\beta s_2$ or $s_0\gamma s_1\beta s_2$ or the others. We introduce a scheduler to convert FDPC into FKS with cost. In this way, the relevant methods in FKS can be used.

Definition 7. Let $M_f = (S, Act, P, I, AP, L, C)$ be a finite FDPC. $Adv : S \rightarrow 2^{Act}$ is a function of M_f . For each $s \in S$, there is $Adv(s) \subseteq Act(s)$.

Under the scheduler Adv , $Path_{Adv}(s)$ denotes the set of infinite paths which start from the state s , $Path_{Adv}^{fin}(M_f)$ denotes the set of finite paths which start from all initial states in M_f . $Path_{Adv}(M_f)$ denotes the set of infinite paths which start from all initial states in M_f .

We often care about the maximum (or minimum) possibility. We select the maximum (or minimum) possibility of transition from state s to t by action in $Adv(s)$ as an example to introduce our thought of scheduler. If there are two or more actions in $Adv(s)$ such that $P(s, \alpha_j, t) = P(s, \alpha_k, t) = \bigvee_{\alpha_i \in Adv(s)} P(s, \alpha_i, t)$ for $\alpha_j, \alpha_k, \alpha_i \in Adv(s)$, we can select the action by the algebraic product $P(s, \alpha_i, t) \cdot C(s, \alpha_i)$. Through the operation of a scheduler, an FDPC can be switched to an FKS with cost.

Remark 1. It is easy to prove that the select operations do not change the maximum or minimum possibility of the Adv , because we can use the actions which are selected by us to replace the actions in the maximum or minimum possibility path.

Example 2. For the FDPC of Figure 4, suppose the scheduler function is $Adv(s_0) = \{\alpha, \beta\}$, $Adv(s_1) = \{\beta, \gamma\}$, $Adv(s_2) = \{\gamma\}$. The FDPC of Adv , the maximum possibility transition FDPC of Adv and the minimum possibility transition FDPC of Adv are shown in Figures 5–7. In fact, Figures 6 and 7 are FKSs with cost.

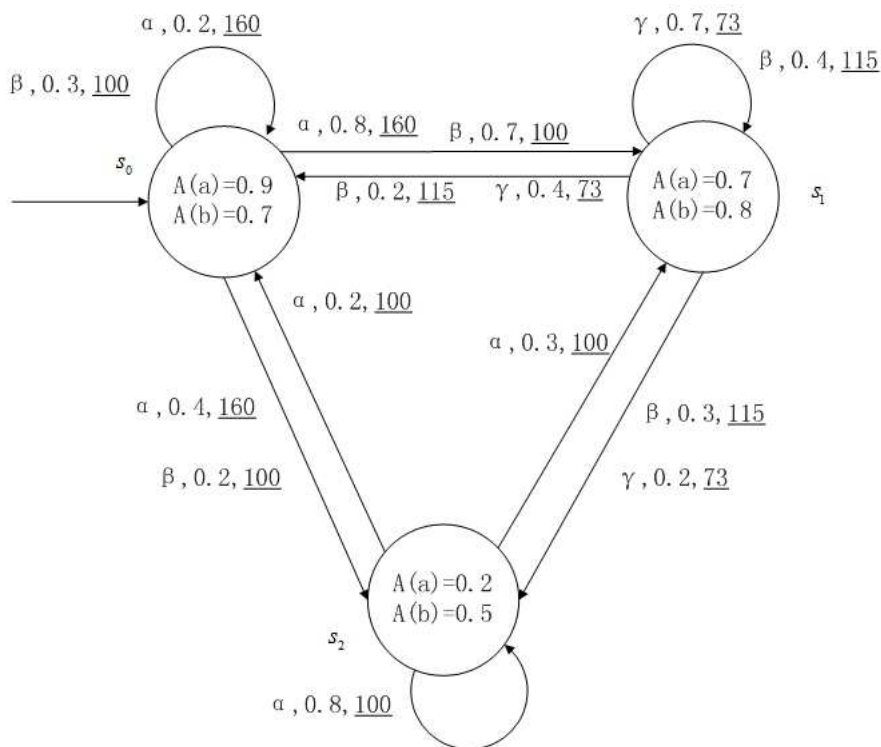


Figure 5. FDPC of Adv .

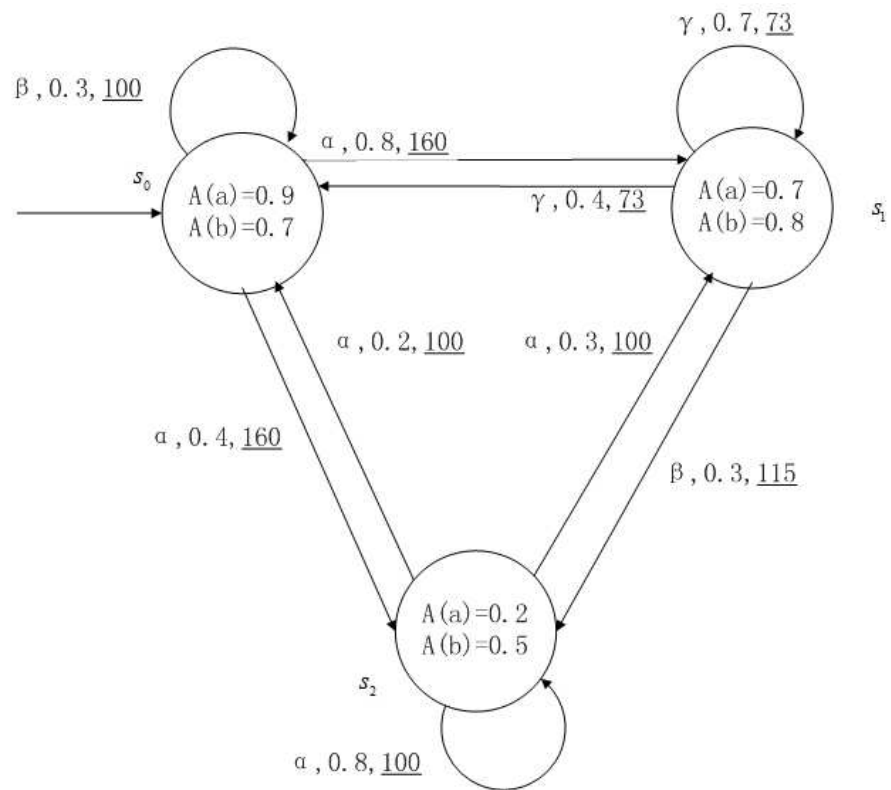


Figure 6. The maximum possibility transition FDPC of Adv.

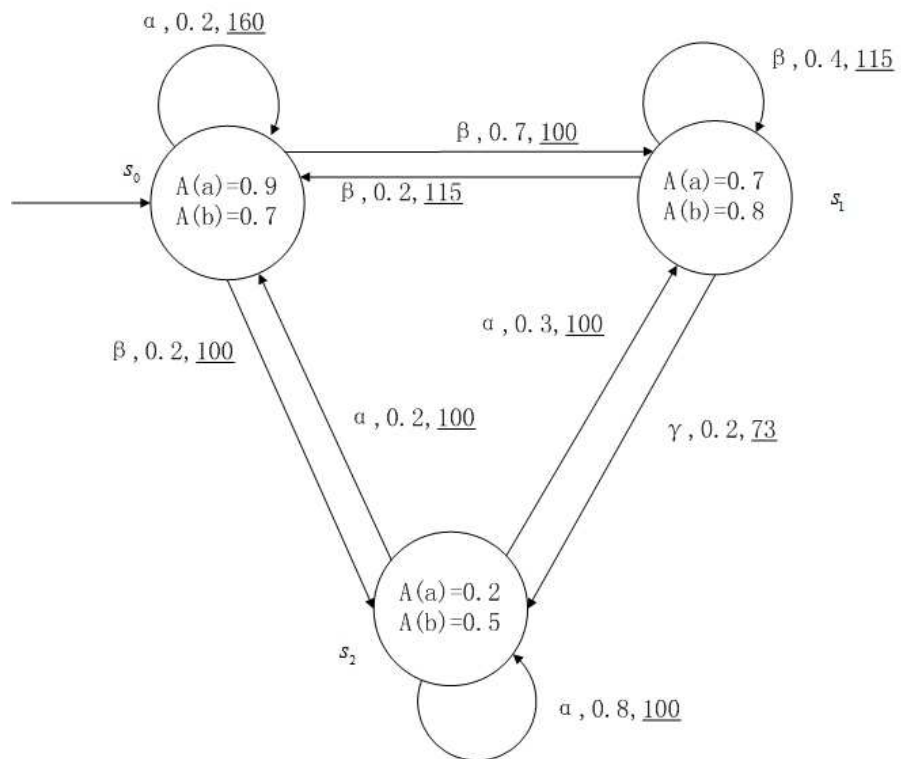


Figure 7. The minimum possibility transition FDPC of Adv.

The actions are eliminated in the conversion period, so we design an action index matrix to store those actions. The transition matrix in the corresponding FKS and the index matrix for recording actions under a specific scheduler are given below.

Let $M_f = (S, Act, P, I, AP, L, C)$ be a finite FDPC and $\pi = s_0\alpha_0s_1\alpha_1s_2... \in Paths(s)$ be a path of M_f .

P_α is a $|S| \times |S|$ fuzzy matrix of transition possibility under the action α . For each $s, t \in S$,

$$P_\alpha(s, t) \stackrel{def}{=} P(s, \alpha, t).$$

The left of the equation is the direct transition which transmit from s to t in FKS which is transmitted by α with cost, but the right is the direct transition which transmit from s to t by act α in FDPC.

$P_{Adv-max}$ is a $|S| \times |S|$ fuzzy matrix which denotes the matrix of maximum transition possibility of Adv . For each $s, t \in S$,

$$P_{Adv-max}(s, t) \stackrel{def}{=} \bigvee_{\alpha \in Adv(s)} P(s, \alpha, t).$$

$T_{Adv-max}$ is a $|S| \times |S|$ action index matrix which records the actions creating the maximum transition possibility. For each $s, t \in S$,

$$T_{Adv-max}(s, t) \stackrel{def}{=} \underset{\alpha \in Adv(s)}{argmax} P(s, \alpha, t).$$

$P_{Adv-min}$ is a $|S| \times |S|$ fuzzy matrix which denotes the matrix of minimum transition possibility of Adv . For each $s, t \in S$,

$$P_{Adv-min}(s, t) \stackrel{def}{=} \bigwedge_{\alpha \in Adv(s)} P(s, \alpha, t).$$

$T_{Adv-min}$ is a $|S| \times |S|$ action index matrix which records the actions creating the minimum transition possibility. For each $s, t \in S$,

$$T_{Adv-min}(s, t) \stackrel{def}{=} \underset{\alpha \in Adv(s)}{argmin} P(s, \alpha, t).$$

We often pay attention to the maximum and minimum possibility of FDPC, but they are the special Adv where $Adv(s) \equiv Act(s)$ for all $s \in S$. We use the special symbol $\alpha_{max}, \alpha_{min}$ to denote the index of this Adv .

$P_{\alpha_{max}}$ is a $|S| \times |S|$ fuzzy matrix which denotes the matrix of maximum transition possibility of FDPC. For each $s, t \in S$,

$$P_{\alpha_{max}}(s, t) \stackrel{def}{=} \bigvee_{\alpha \in Act(s)} P(s, \alpha, t).$$

$T_{\alpha_{max}}$ is a $|S| \times |S|$ action index matrix which records the actions creating the maximum transition possibility. For each $s, t \in S$,

$$T_{\alpha_{max}}(s, t) \stackrel{def}{=} \underset{\alpha \in Act(s)}{argmax} P(s, \alpha, t).$$

$P_{\alpha_{min}}$ is a $|S| \times |S|$ fuzzy matrix which denotes the matrix of minimum transition possibility of FDPC. For each $s, t \in S$,

$$P_{\alpha_{min}}(s, t) \stackrel{def}{=} \bigwedge_{\alpha \in Act(s)} P(s, \alpha, t).$$

$T_{\alpha_{min}}$ is a $|S| \times |S|$ action index matrix which records the actions creating the minimum transition possibility. For each $s, t \in S$,

$$T_{\alpha_{min}}(s, t) \stackrel{def}{=} \underset{\alpha \in Act(s)}{argmin} P(s, \alpha, t).$$

C_α is a $|S| \times 1$ matrix declaring the cost activating action α from state s , for each $s \in S$,

$$C_\alpha \stackrel{def}{=} C(s, \alpha).$$

Example 3. For the FDPC of Figure 4.

The action α transition possibility matrix and the cost matrix activating action α is

$$P_\alpha = \begin{pmatrix} 0.2 & 0.8 & 0.4 \\ 0.1 & 0.3 & 0.5 \\ 0.2 & 0.3 & 0.8 \end{pmatrix}, C_\alpha = \begin{pmatrix} 160 \\ 180 \\ 100 \end{pmatrix}.$$

The action β transition possibility matrix and the cost matrix activating action β is

$$P_\beta = \begin{pmatrix} 0.3 & 0.7 & 0.2 \\ 0.2 & 0.4 & 0.3 \\ 0.1 & 0.5 & 0.4 \end{pmatrix}, C_\beta = \begin{pmatrix} 100 \\ 115 \\ 70 \end{pmatrix}.$$

The action γ transition possibility matrix and the cost matrix activating action γ is

$$P_\gamma = \begin{pmatrix} 0.5 & 0.6 & 0.1 \\ 0.4 & 0.7 & 0.2 \\ 0.5 & 0.6 & 0.3 \end{pmatrix}, C_\gamma = \begin{pmatrix} 68 \\ 73 \\ 40 \end{pmatrix}.$$

The maximum transition possibility matrix, maximum transition possibility action index matrix, minimum transition possibility matrix and minimum transition possibility action index matrix is

$$P_{\alpha_{max}} = \begin{pmatrix} 0.5 & 0.8 & 0.4 \\ 0.4 & 0.7 & 0.5 \\ 0.5 & 0.6 & 0.8 \end{pmatrix}, T_{\alpha_{max}} = \begin{pmatrix} \gamma & \alpha & \alpha \\ \gamma & \gamma & \alpha \\ \gamma & \gamma & \alpha \end{pmatrix},$$

$$P_{\alpha_{min}} = \begin{pmatrix} 0.2 & 0.6 & 0.1 \\ 0.1 & 0.3 & 0.2 \\ 0.1 & 0.3 & 0.3 \end{pmatrix}, T_{\alpha_{min}} = \begin{pmatrix} \alpha & \gamma & \gamma \\ \alpha & \alpha & \gamma \\ \beta & \alpha & \gamma \end{pmatrix}.$$

Under the maximum transition possibility matrix and minimum transition possibility matrix, the FDPC of Figure 4 turns into the FKSs with cost in Figures 8 and 9.

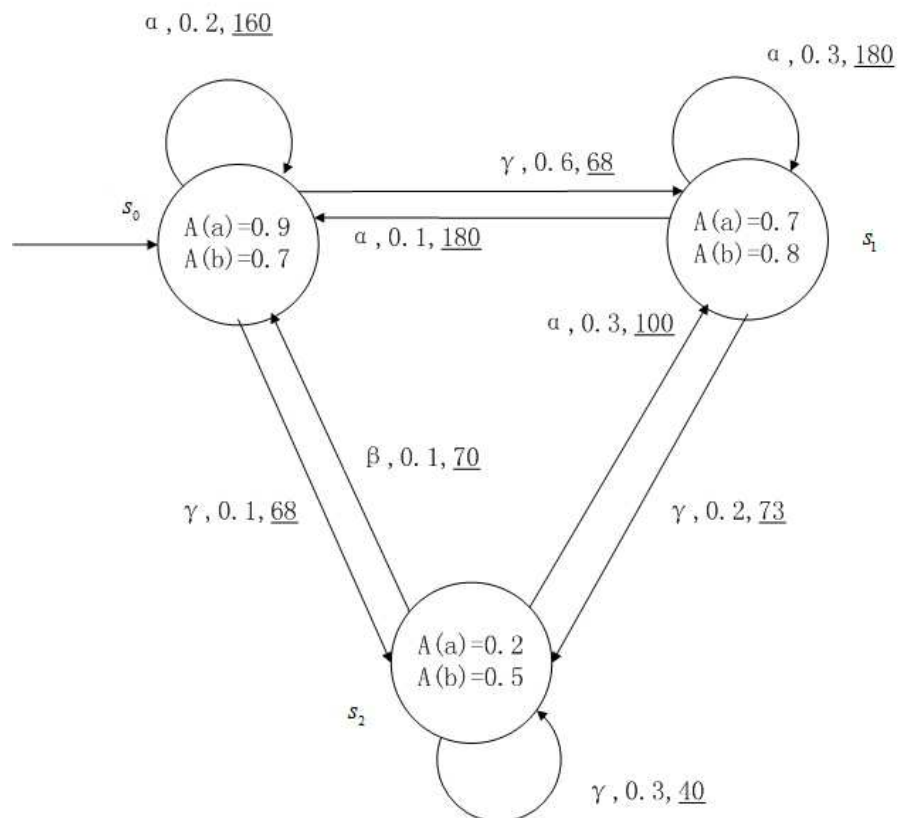


Figure 8. FDPC of $P_{\alpha_{max}}$.

Since the cost is generated in the process of each activation action, in order to solve the cost-related problems in model checking, we determine the cost through a deterministic selection strategy for the action, and then calculate the expected cost. Because model checking pays more attention to the possibility of transition, this paper takes the maximum possibility of one-step transition as the selected strategy. First, we select the action by the above matrix. Then, we select the successor state by the maximum possibility.

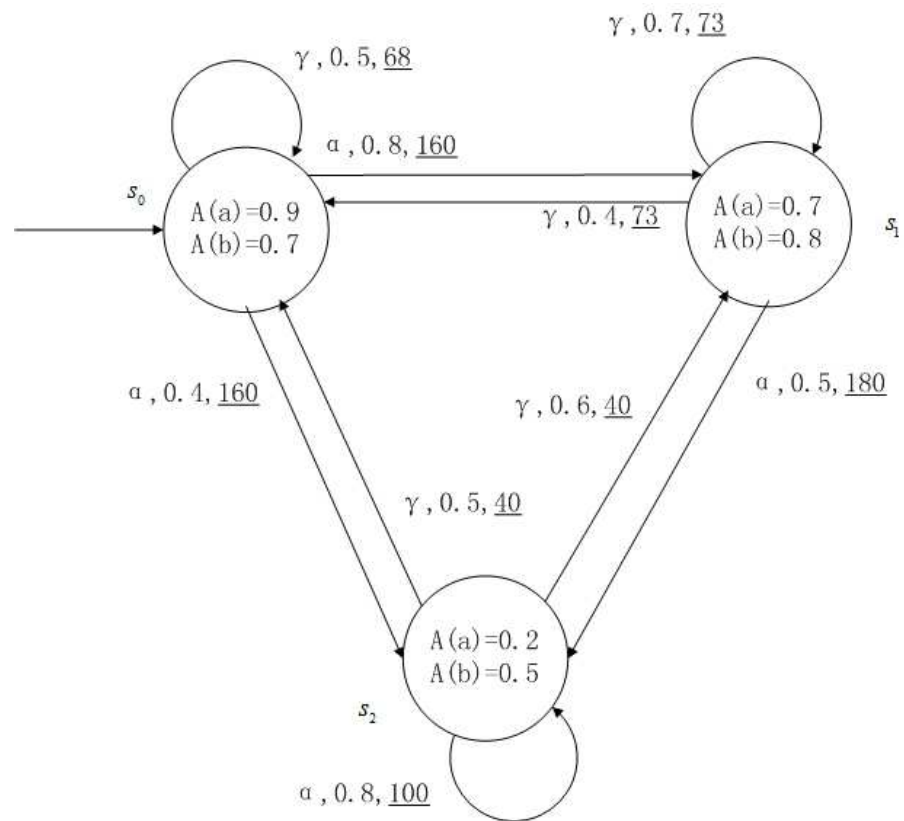


Figure 9. FDPC of $P_{\alpha_{min}}$.

Definition 8. Let $M_f = (S, Act, P, I, AP, L, C)$ be a finite FDPC and $\pi = s_0\alpha_0s_1\alpha_1s_2\dots \in Paths(s)$ be a path of M_f . $cost[= k](\pi) \stackrel{def}{=} C(s_{k-1}, \alpha_{k-1})$ denotes the instantaneous cost of the step k .

Under the scheduler Adv , we use each step to choose the maximum possibility to transmit from the current state as an example, and the step k instantaneous expected cost is defined as

$$Ex_{Adv}^s(cost[= k]) \stackrel{def}{=} \left(\bigwedge_{0 < j \leq k} \left(\bigvee_{t_j \in S} P_{Adv}(t_{j-1}, t_j) \right) \right) \cdot C(t_{j-1}, \alpha_{k-1}).$$

We use $cost[\leq k](\pi) \stackrel{def}{=} \sum_{i=1}^k C(s_{i-1}, \alpha_{i-1})$ to denote the cumulative cost of the first k steps. The cumulative expected cost of the first k steps is defined as

$$Ex_{Adv}^s(cost[\leq k]) \stackrel{def}{=} \sum_{i=1}^k Ex_{Adv}^s(cost[= i]).$$

The previous descriptions are all about the expected cost without limiting the states in the path. However, in the actual process, some restrictions may be added to the states in the path.

$$cost[\diamond F](\pi) = cost[\wedge] \stackrel{def}{=} \sum_{i=1}^n C(s_{i-1}, \alpha_{i-1})$$

denotes the cumulative cost that the path would reach the state in F , where $F \subseteq S$, $\wedge = s_0\alpha_0\dots s_n$ is the prefix of π and $s_n \in F$ but the other states in \wedge is not in F .

$Ex_{Adv}^s(cost[= n(\diamond F)])$ denotes the skip n instantaneous expected cost under the scheduler Adv which is the deformation of $Ex_{Adv}^s(cost[= k])$, defined as below.

$$Ex_{Adv}^s(cost[= 1(\diamond F)]) \stackrel{def}{=} \left(\bigvee_{t \in F} P_{Adv}(s, t) \right) \cdot C(s, \alpha),$$

$$Ex_{Adv}^s(cost[=k(\diamond F)]) \stackrel{def}{=} \left(\bigwedge_{0 < j < k} \left(\bigvee_{t \in S/F} P_{Adv}(t_{j-1}, t_j) \right) \wedge P_{Adv}(t_k - 1, t) \right) \cdot C(t_{j-1}, \alpha_{k-1}).$$

$Ex_{Adv}^s(cost[\leq k(\diamond F)])$ denotes the first steps k cumulative expected cost under the scheduler Adv which reaches F in step k , defined as below,

$$\begin{aligned} Ex_{Adv}^s(cost[\leq 1(\diamond F)]) &\stackrel{def}{=} Ex_{Adv}^s(cost[=1(\diamond F)]), \\ Ex_{Adv}^s(cost[\leq k(\diamond F)]) &\stackrel{def}{=} \\ &\left(\sum_{i=1}^{k-1} \left(\bigvee_{t \in F} \left(\bigwedge_{0 < j < k} \left(\bigvee_{s \in S/F} P_{Adv}(t_{j-1}, t_j) \right) \right) \right) \right) \cdot \\ &C(s_{j-1}, \alpha_{j-1}) + \left(\bigvee_{t \in F} \left(\bigwedge_{0 < j < k} \left(\bigvee_{s \in S/F} P_{Adv}(t_{j-1}, t_j) \wedge P_{Adv}(s_{k-1}, t) \right) \right) \right) \cdot C(s_{k-1}, \alpha_{k-1}). \end{aligned}$$

$Ex_{Adv}^s(cost[\diamond F])$ is the cumulative expected cost from s to the state in F , defined as below,

$$Ex_{Adv}^s(cost[\diamond F]) \stackrel{def}{=} \sup_{1 \leq k \leq n} Ex_{Adv}^s(cost[\leq k(\diamond F)]).$$

Where n is the max step of all paths that can reach F under the restrictive condition 'No or one ring'. Why do we have the restrictive condition 'No or one ring'? Because it has no contribution for transition possibility changing. Without the amount of rings, all states of the path which can reach F are less than or equal $|S/F| + 1$, thus we can get that the max skep $n \leq |S/F|$.

4. FCTL with Cost Operator

We present the FCTL with cost operators in this section, i.e., expand FCTL for our FDPC with cost operator. We expand FCTL [19] with the cost operator. The syntax and semantics are as below.

Definition 9 (FCTL syntax). *The FCTL state formula is defined inductively as follows,*

$$\Phi ::= true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \exists \varphi \mid \forall \varphi \mid \mathbb{E}(=k) \mid \mathbb{E}(\leq k) \mid \mathbb{E}(\Phi),$$

where φ is a path formula, $a \in AP$.

Furthermore, the FCTL path formula is,

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

where Φ, Φ_1 and Φ_2 are state formulas.

Definition 10 (FCTL semantics). *Let $M_f = (S, Act, P, I, AP, L, C)$ be an finite FDPC, $\|\Phi\|: S \rightarrow [0, 1]$ be a function. For FCTL with cost, the semantic of state formula Φ is defined as follows.*

$$\|\text{true}\| (s) = 1,$$

$$\|a\| (s) = L(s, a),$$

$$\|\Phi_1 \wedge \Phi_2\| (s) = \|\Phi_1\| (s) \wedge \|\Phi_2\| (s),$$

$$\|\neg \Phi\| (s) = 1 - \|\Phi\| (s),$$

$$\|\exists \varphi\|_{Adv} (s) = \bigvee_{\pi \in Paths_{Adv}(s)} \|\varphi\| (\pi),$$

$$\|\forall \varphi\|_{Adv} (s) = \bigwedge_{\pi \in Paths_{Adv}(s)} \|\varphi\| (\pi),$$

$$\|\mathbb{E}(=k)\|_{Adv} (s) = Ex_{Adv}^s(cost[=k]),$$

$$\|\mathbb{E}(\leq k)\|_{Adv} (s) = Ex_{Adv}^s(cost[\leq k]),$$

$$\|\mathbb{E}(\Phi)\|_{Adv} (s) = Ex_{Adv}^s(cost[\diamond Sat(\Phi)])$$

where $Sat(\Phi) = \{s \in S \mid \|\Phi\| (s) > 0\}$.

For the given scheduler Adv and $\pi \in Paths_{Adv}(s)$, the semantic of path formula φ is defined as below.

$$\|\bigcirc \Phi\|_{Adv} = P_{Adv}(s_0, s_1) \wedge \|\Phi\| (s_1),$$

$$\|\Phi_1 \sqcup \Phi_2\|_{Adv} (\pi)$$

$$= \|\Phi_2\| (s) \vee \bigvee_{j>0} (\|\Phi_1\| (s_0) \wedge \bigwedge_{k<j} (P_{Adv}(s_{k-1}, s_k)) \wedge \|\Phi_1\| (s_k) \wedge (P_{Adv}(s_{j-1}, s_j)) \wedge$$

$$\|\Phi_2\| (s_j)).$$

5. Model Checking Fuzzy Computation Tree Logic Based on Fuzzy Decision Processes with Cost

The FCTL model checking problem on FDPC is defined as the following. Given a FDPC M_f , a state s of M_f , a FCTL state formula Φ and a step k , then calculate the true value of state s satisfying the state formula or the expect cost. We often consider the maximum and minimum possibilistic truth values. We give the computing method of $\|\exists \bigcirc \Phi\|_{max}(s), \|\exists \bigcirc \Phi\|_{min}(s), \|\forall \bigcirc \Phi\|_{max}(s), \|\forall \bigcirc \Phi\|_{min}(s), \|\exists \Phi_1 \sqcup \Phi_2\|_{max}(s), \|\exists \Phi_1 \sqcup \Phi_2\|_{min}(s), \|\forall \Phi_1 \sqcup \Phi_2\|_{max}(s), \|\forall \Phi_1 \sqcup \Phi_2\|_{min}(s), \|\mathbb{E}(=k)\|_{max}(s), \|\mathbb{E}(=k)\|_{min}(s), \|\mathbb{E}(\leq k)\|_{max}(s), \|\mathbb{E}(\leq k)\|_{min}(s), \|\mathbb{E}(\Phi)\|_{max}(s)$ and $\|\mathbb{E}(\Phi)\|_{min}(s)$ as below. There are some useful matrixes and operations given.

Let $M_f = (S, Act, P, I, AP, L, C)$ be a finite FDPC, D_Φ be a $|S| \times |S|$ fuzzy diagonal matrix for state formula Φ . For each $s, t \in S$,

$$D_\Phi(s, t) \stackrel{def}{=} \begin{cases} \|\Phi\|(s) & s = t \\ 0 & \text{Otherwise} \end{cases}$$

P_Φ is a $|S| \times 1$ fuzzy matrix. For each $s \in S$,

$$P_\Phi(s) \stackrel{def}{=} \|\Phi\|(s).$$

We use $P_{\alpha|_{max}}$ to realize the transition of only choosing the maximum truth value transition using action α . $P_{\alpha|_{max}}$ is a $|S| \times |S|$ fuzzy matrix and defined as that for each $s, t \in S$,

$$P_{\alpha|_{max}}(s, t) \stackrel{def}{=} \begin{cases} P(s, \alpha, t) & t = \underset{t \in S}{argmax} f(P(s, \alpha, t), cost(s, \alpha)) \\ 0 & \text{Otherwise} \end{cases}$$

where f is a function mapping $P \times C$ to $[0,1]$. f is decided by the need in usual. In this paper, we set $f(P(s, \alpha, t), cost(s, \alpha)) = P(s, \alpha, t)$ if there is only one maximum transition from s to t . When it is multiple, we select the first maximum transition by elements in the matrix.

Through those matrixes, we can reduce the state transition matrix in FKS to a sparse matrix containing only one-step maximum possibility.

E is a $|S| \times 1$ fuzzy matrix with all elements equal to 1. We use it to turn the matrix into a vector.

We also use an auxiliary matrix identified as D_F , for the restricted set F . D_F is defined below.

$$D_F = \begin{pmatrix} D[0,0] & \cdots & D[0,n] \\ \dots & \vdots & \dots \\ D[n,0] & \cdots & D[n,n] \end{pmatrix},$$

where $D[i, j] = \begin{cases} 1 & s_j \in F \\ 0 & s_j \notin F \end{cases}$

$P_{\alpha|_F} \stackrel{def}{=} D_F \cap P_\alpha$ is the restricted transition matrix under the act α which restrict only transition to F . $P'_{\alpha|_F} \stackrel{def}{=} (D_S - D_F) \cap P_\alpha$ is the restricted transition matrix under the act α which cannot transmit to F . Using matrixes, we can describe the constraints.

Using our matrixes, we can re-represent the cost in Definition 8.

The first k steps cumulative expected cost

$$\begin{aligned} & Ex_{Adv}^s(cost[\leq k]) \\ &= \sum_{i=1}^k Ex_{Adv}^s(cost[= i]) \\ &= \sum_{i=1}^k ((\bigwedge_{0 < j \leq i} (\bigvee_{t_j \in S} P_{Adv}(t_{j-1}, t_j))) \cdot C(t_{i-1}, \alpha_{i-1})) \\ &= \sum_{i=1}^k ((\bigwedge_{0 < j \leq i} (\bigvee_{t_j \in S} \bigvee_{\alpha_{j-1} \in Adv(t_{j-1})} P_{Adv}(t_{j-1}, \alpha_{j-1}, t_j))) \cdot C(t_{i-1}, \alpha_{i-1})) \\ &= \sum_{i=1}^k ((\bigwedge_{0 < j \leq i} (\bigvee_{t_j \in S} P_{Adv-max}(t_{j-1}, t_j))) \cdot C(t_{i-1}, \alpha_{i-1})) \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^k ((P_{Adv-max|max|} \circ P_{Adv-max|max|} \circ \dots \circ P_{Adv-max|max|} \circ E)(s) \cdot C_{\alpha_{i-1}}(t_{i-1})) \\ &= \sum_{i=1}^k ((P_{Adv-max|max|}^i \circ E)(s) \cdot C_{\alpha_{i-1}}(t_{i-1})). \end{aligned}$$

The restricted set $F \subseteq S$ step k instantaneous expected cost

$$\begin{aligned} Ex_{Adv}^s(cost[= 1(\diamond F)]) &= (\bigvee_{t \in F} P_{Adv}(s, t)) \cdot C(s, \alpha) \\ &= (\bigvee_{t \in F} \bigvee_{\alpha \in Adv(s)} P(s, \alpha, t)) \cdot C(s, \alpha) \\ &= ((P_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha}(s)) \\ Ex_{Adv}^s(cost[= k(\diamond F)]) &= (\bigvee_{t \in F} (\bigwedge_{0 < j < k} (\bigvee_{s \in S \setminus F} P_{Adv}(s_{j-1}, s_j)) \wedge P_{Adv}(s_{k-1}, t))) \cdot C(s_{k-1}, \alpha_{k-1}) \\ &= (\bigvee_{t \in F} (\bigwedge_{0 < j < k} (\bigvee_{s \in S \setminus F} \bigvee_{\alpha_{j-1} \in Adv(s_{j-1})} P(s_{j-1}, \alpha_{j-1}, s_j)) \wedge \bigvee_{\alpha_{k-1} \in Adv(s_{k-1})} P(s_{k-1}, \alpha_{k-1}, t))) \cdot C(s_{k-1}, \alpha_{k-1}) \\ &= ((P'_{Adv-max|F|max|} \circ \dots \circ P'_{Adv-max|F|max|} \circ P_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha_{k-1}}(s_{k-1})) \\ &= ((P'^{k-1}_{Adv-max|F|max|} \circ P_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha_{k-1}}(s_{k-1})) \end{aligned}$$

The restricted set $F \subseteq S$ first k steps cumulative expected cost

$$\begin{aligned} Ex_{Adv}^s(cost[\leq 1(\diamond F)]) &= Ex_{Adv}^s(cost[= 1(\diamond F)]) \\ &= (P_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha_{k-1}}(s) \\ Ex_{Adv}^s(cost[\leq k(\diamond F)]) &= \sum_{i=1}^{k-1} ((\bigwedge_{0 < j \leq i} (\bigvee_{s \in S \setminus F} P_{Adv}(s_{j-1}, s_j))) \cdot C(s_{j-1}, \alpha_{j-1})) + (\bigvee_{t \in F} (\bigwedge_{0 < j < k} (\bigvee_{s \in S \setminus F} P_{Adv}(s_{j-1}, s_j)) \wedge P_{Adv}(s_{k-1}, t))) \cdot C(s_{k-1}, \alpha_{k-1}) \\ &= \sum_{i=1}^{k-1} ((P'^i_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha_{i-1}}(s_{i-1})) + ((P'^{k-1}_{Adv-max|F|max|} \circ P_{Adv-max|F|max|} \circ E)(s) \cdot C_{\alpha_{k-1}}(s_{k-1})) \end{aligned}$$

where $P_{\alpha|F|max|}$ is the operation that to P_{α} first count $P_{\alpha|F}$ and second count $P_{\alpha|max|}$ for $P_{\alpha|F}$. $P'_{\alpha|F|max|}$ is the operation that to P_{α} first count $P'_{\alpha|F}$ and second count $P_{\alpha|max|}$ for $P'_{\alpha|F}$. $\|\exists \circ \Phi\|_{max}(s)$ is the maximum truth value of that there exists a path that starts from state s and satisfies $\circ \Phi$.

$$\|\exists \circ \Phi\|_{max}(s) = (P_{\alpha_{max}} \circ P_{\Phi})(s). \tag{1}$$

The proof is placed in Appendix A.

$\|\exists \circ \Phi\|_{min}(s)$ is the minimum truth value of that there exists a path that starts from state s and satisfies $\circ \Phi$.

$$\|\exists \circ \Phi\|_{min}(s) = (P_{\alpha_{min}} \circ P_{\Phi})(s). \tag{2}$$

The proof is placed in Appendix A.

$\|\forall \circ \Phi\|_{max}(s)$ is the maximum truth value of that all of the paths which start from state s satisfy $\circ \Phi$.

$$\|\forall \circ \Phi\|_{max}(s) = ((P_{\alpha_{max}} \circ D_{\Phi})^c \circ E)^c(s) \tag{3}$$

The proof is placed in Appendix A.

$\|\forall \circ \Phi\|_{min}(s)$ is the minimum truth value of that all of the paths which start from state s satisfy $\circ \Phi$.

$$\|\forall \circ \Phi\|_{min}(s) = ((P_{\alpha_{min}} \circ D_{\Phi})^c \circ E)^c(s) \tag{4}$$

The proof is placed in Appendix A.

$\|\exists\Phi_1 \sqcup \Phi_2\|_{max}(s)$ is the maximum truth value of that there exists a path that starts from state s satisfies $\Phi_1 \sqcup \Phi_2$.

$$\|\exists\Phi_1 \sqcup \Phi_2\|_{max}(s) = ((D_{\Phi_1} \circ P_{\alpha_{max}})^* \circ P_{\Phi_2})(s) \tag{5}$$

The proof is placed in Appendix A.

$\|\exists\Phi_1 \sqcup \Phi_2\|_{min}(s)$ is the minimum truth value of that there exists a path that starts from state s satisfies $\Phi_1 \sqcup \Phi_2$.

$$\|\exists\Phi_1 \sqcup \Phi_2\|_{min}(s) = ((D_{\Phi_1} \circ P_{\alpha_{min}})^* \circ P_{\Phi_2})(s) \tag{6}$$

The proof is placed in Appendix A.

$\|\forall\Phi_1 \sqcup \Phi_2\|_{max}(s)$ is the maximum truth value of that all of the paths which start from state s satisfy $\Phi_1 \sqcup \Phi_2$.

$$\begin{aligned} &\|\forall\Phi_1 \sqcup \Phi_2\|_{max}(s) \\ &= (P_{\Phi_2} \cup (((D_{\Phi_1} \circ P_{\alpha_{max}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{max}})^c)^* \circ D_{\Phi_2})^c \\ &\quad \circ E)^c)(s) \end{aligned} \tag{7}$$

The proof is placed in Appendix A.

$\|\forall\Phi_1 \sqcup \Phi_2\|_{min}(s)$ is minimum the truth values of that all of the paths that start from state s satisfy $\Phi_1 \sqcup \Phi_2$.

$$\begin{aligned} &\|\forall\Phi_1 \sqcup \Phi_2\|_{min}(s) \\ &= (P_{\Phi_2} \cup (((D_{\Phi_1} \circ P_{\alpha_{min}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{min}})^c)^* \circ D_{\Phi_2})^c \\ &\quad \circ E)^c)(s) \end{aligned} \tag{8}$$

The proof is placed in Appendix A.

$\|\mathbb{E}(=k)\|_{max}(s)$ is the skip k instantaneous expected cost of the path that starts from state s under the maximum scheduler.

$$\begin{aligned} &\|\mathbb{E}(=0)\|_{max}(s) = 0, \\ &\|\mathbb{E}(=k)\|_{max}(s) = (P_{\alpha_{max}|max}^k \circ E)(s) \cdot C_{T_{\alpha_{max}}(t_{k-1}, t_k)}(t_{k-1}) \end{aligned} \tag{9}$$

where $t_m = \underset{t_m \in S}{argmax}(P_{\alpha_{max}|max}^m(s, t_m))$.

The proof is placed in Appendix A.

$\|\mathbb{E}(=k)\|_{min}(s)$ is the skip k instantaneous expected cost of the path that starts from state s under the minimum scheduler.

$$\begin{aligned} &\|\mathbb{E}(=0)\|_{min}(s) = 0, \\ &\|\mathbb{E}(=k)\|_{min}(s) = (P_{\alpha_{min}|max}^k \circ E)(s) \cdot C_{T_{\alpha_{min}}(t_{k-1}, t_k)}(t_{k-1}) \end{aligned} \tag{10}$$

where $t_m = \underset{t_m \in S}{argmax}(P_{\alpha_{min}|max}^m(s, t_m))$.

The proof is placed in Appendix A.

$\|\mathbb{E}(\leq k)\|_{max}(s)$ is the first k steps cumulative expected cost of the path that starts from state s under the maximum scheduler.

$$\begin{aligned}
 & || \mathbb{E}(\leq 0) ||_{max} (s) = Ex_{max}^s(cost[= 0]) = 0 \\
 & || \mathbb{E}(\leq k) ||_{max} (s) \\
 & = Ex_{max}^s(cost[\leq k]) \\
 & = \sum_{i=1}^k ((P_{\alpha_{max}|max}^i \circ E)(s) \cdot C_{\alpha_{i-1}}(t_{i-1})) \\
 & = \sum_{i=1}^k ((P_{\alpha_{max}|max}^i \circ E)(s) \cdot C_{T_{\alpha_{max}}(t_{i-1},t_i)}(t_{i-1})) \\
 & || \mathbb{E}(\leq k) ||_{max} (s) = \sum_{i=1}^k ((P_{\alpha_{max}|max}^i \circ E)(s) \cdot C_{T_{\alpha_{max}}(t_{i-1},t_i)}(t_{i-1})) \tag{11}
 \end{aligned}$$

where $t_m = argmax_{t_m \in S} (P_{\alpha_{max}|max}^m(s, t_m))$.

$|| \mathbb{E}(\leq k) ||_{min} (s)$ is the first k steps cumulative expected cost of the path that starts from state s under the minimum scheduler.

$$\begin{aligned}
 & || \mathbb{E}(\leq 0) ||_{min} (s) = Ex_{min}^s(cost[= 0]) = 0 \\
 & || \mathbb{E}(\leq k) ||_{min} (s) \\
 & = Ex_{min}^s(cost[\leq k]) \\
 & = \sum_{i=1}^k ((P_{\alpha_{min}|max}^i \circ E)(s) \cdot C_{\alpha_{i-1}}(t_{i-1})) \\
 & = \sum_{i=1}^k ((P_{\alpha_{min}|max}^i \circ E)(s) \cdot C_{T_{\alpha_{min}}(t_{i-1},t_i)}(t_{i-1})) \\
 & || \mathbb{E}(\leq k) ||_{min} (s) = \sum_{i=1}^k ((P_{\alpha_{min}|max}^i \circ E)(s) \cdot C_{T_{\alpha_{min}}(t_{i-1},t_i)}(t_{i-1})) \tag{12}
 \end{aligned}$$

where $t_m = argmax_{t_m \in S} (P_{\alpha_{min}|max}^m(s, t_m))$.

$|| \mathbb{E}(\Phi) ||_{max} (s)$ is the cumulative expected cost of the path that starts from state s and can reach a state in F under the maximum scheduler.

$$\begin{aligned}
 & || \mathbb{E}(\Phi) ||_{max} (s) \\
 & = Ex_{max}^s(cost[\diamond Sat(\Phi)]) \\
 & = \sup_{1 \leq k \leq |S| - |Sat(\Phi)|} Ex_{max}^s(cost[\leq k(\diamond Sat(\Phi))]) \\
 & = \sup_{2 \leq k \leq |S| - |Sat(\Phi)|} \{Ex_{max}^s(cost[\leq 1(\diamond Sat(\Phi))]), Ex_{max}^s(cost[\leq k(\diamond Sat(\Phi))])\} \\
 & = \sup_{2 \leq k \leq |S| - |Sat(\Phi)|} \{ (P_{\alpha_{max}|Sat(\Phi)|max} \circ E)(s) \cdot C_{T_{\alpha_{max}}(s,t)}(s), \\
 & \sum_{i=1}^{k-1} ((P_{\alpha_{max}|Sat(\Phi)|max}^i \circ E)(s) \cdot C_{T_{\alpha_{max}}(t_{i-1},t_i)}(s)) + \\
 & ((P_{\alpha_{max}|Sat(\Phi)|max}'^{k-1} \circ P_{\alpha_{max}|Sat(\Phi)|max} \circ E)(s) \cdot C_{T_{\alpha_{max}}(t_{k-1},t_k)}(t_{k-1})) \} \tag{13}
 \end{aligned}$$

where $t = argmax_{t \in Sat(\Phi)} (P_{\alpha_{max}|Sat(\Phi)|max}(s, t))$,

$$t_m = argmax_{t_m \in S / Sat(\Phi)} ((P_{\alpha_{max}|Sat(\Phi)|max}'^m(s, t_m)), \text{ if } 1 \leq m \leq k - 1,$$

$$t_m = argmax_{t_m \in Sat(\Phi)} (((P_{\alpha_{max}|Sat(\Phi)|max}'^m \circ P_{\alpha_{max}|Sat(\Phi)|max})(s, t_m)) \text{ if } m = k.$$

$\|\mathbb{E}(\Phi)\|_{min}(s)$ is the cumulative expected cost of the path that starts from state s and can reach a state in F under the minimum scheduler.

$$\begin{aligned}
 & \|\mathbb{E}(\Phi)\|_{min}(s) \\
 &= Ex_{min}^s(cost[\diamond Sat(\Phi)]) \\
 &= \sup_{1 \leq k \leq |S| - |Sat(\Phi)|} Ex_{min}^s(cost[\leq k(\diamond Sat(\Phi))]) \\
 &= \sup_{2 \leq k \leq |S| - |Sat(\Phi)|} \{Ex_{min}^s(cost[\leq 1(\diamond Sat(\Phi))]), \\
 & Ex_{min}^s(cost[\leq k(\diamond Sat(\Phi))])\} \\
 &= \sup_{2 \leq k \leq |S| - |Sat(\Phi)|} \{(P_{\alpha_{min}|Sat(\Phi)|max} \circ E)(s) \cdot C_{T_{\alpha_{min}}(s,t)}(s), \\
 & \sum_{i=1}^{k-1} ((P'_{\alpha_{min}|Sat(\Phi)|max})^i \circ E)(s) \cdot C_{T_{\alpha_{min}}(t_{i-1},t_i)}(s)) + \\
 & ((P'_{\alpha_{min}|Sat(\Phi)|max})^{k-1} \circ P_{\alpha_{min}|Sat(\Phi)|max} \circ E)(s) \cdot C_{T_{\alpha_{min}}(t_{k-1},t_k)}(t_{k-1})\} \tag{14}
 \end{aligned}$$

where $t = \underset{t \in Sat(\Phi)}{argmax}(P_{\alpha_{min}|Sat(\Phi)|max}(s, t))$,

$$t_m = \underset{t_m \in S/Sat(\Phi)}{argmax} ((P'_{\alpha_{min}|Sat(\Phi)|max})^m(s, t_m)), \text{ if } 1 \leq m \leq k - 1,$$

$$t_m = \underset{t_m \in Sat(\Phi)}{argmax} (((P'_{\alpha_{min}|Sat(\Phi)|max})^m \circ P_{\alpha_{min}|Sat(\Phi)|max})(s, t_m)) \text{ if } m = k.$$

According to (1)–(14), we provide three algorithms to solve the problem of FCTL model checking with cost. Algorithm 1 is used to catch some values of some parameters which would be used to calculate the cost operators. Algorithm 2 is used to calculate the truth values of the formal FCTL state formulas. Algorithm 3 is used to calculate the cost operators.

Algorithm 1 Catch the action

Require: a state s , the first $k - 1$ step transition matrix P_{α} , the step k transition matrix P_{β} , action index matrix T_{α} .

Ensure: the state t_{k-1} after $k - 1$ steps transition, the state t_k after k steps transition, the action of step k $T_{\alpha}(t_{k-1}, t_k)$.

- 1: **for** $t \in S$ **do**
 - 2: **if** $P_{\alpha}^{k-1}(s, t) > 0$ **then**
 - 3: $t_{k-1} \leftarrow t$
 - 4: **end if**
 - 5: **if** $(P_{\alpha}^{k-1} \circ P_{\beta})(s, t) > 0$ **then**
 - 6: $t_k \leftarrow t$
 - 7: **end if**
 - 8: **end for**
 - 9: **return** $t_{k-1}, t_k, T_{\alpha}(t_{k-1}, t_k)$
-

Algorithm 1 is proposed to get the action α and states s_{k-1} and s_k in transition $s_{k-1} \xrightarrow{\alpha} s_k$ which are used in the computing of cost operators. By the definition of $P_{\alpha|max}$, we can use the $P_{\alpha}^{k-1}(s, t) > 0$ to be the determined condition of which is the successor state.

Algorithm 2 Calculating the formal FCTL state formula**Require:** a FDPC M_f , a FCTL state formula Φ .**Ensure:** the truth value of $\|\Phi\|(s)$.

```

1: if  $\Phi = true$  then
2:   return  $(1)_{s \in S}$ 
3: end if
4: if  $\Phi = a \in AP$  then
5:   return  $(\|a\|(s))_{s \in S}$ 
6: end if
7: if  $\Phi = \neg\Phi$  then
8:   return  $(1 - \|\Phi\|(s))_{s \in S}$ 
9: end if
10: if  $\Phi = \Phi_1 \wedge \Phi_2$  then
11:   return  $(\|\Phi_1\|(s) \wedge \|\Phi_2\|(s))_{s \in S}$ 
12: end if
13: if  $\Phi = \exists \bigcirc \Phi$  then
14:   return  $P_{Adv} \circ P_\Phi$ 
15: end if
16: if  $\Phi = \forall \bigcirc \Phi$  then
17:   return  $((P_{Adv} \circ D_\Phi)^c \circ E)^c$ 
18: end if
19: if  $\Phi = \exists \Phi_1 \sqcup \Phi_2$  then
20:   return  $(D_{\Phi_1} \circ P_{Adv})^* \circ P_{\Phi_2}$ 
21: end if
22: if  $\Phi = \forall \Phi_1 \sqcup \Phi_2$  then
23:   return  $P_{\Phi_2} \cup (((D_{\Phi_1} \circ P_{Adv})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{Adv})^c)^c)^* \circ D_{\Phi_2})^c \circ E)^c$ 
24: end if

```

Algorithm 2 is proposed to calculate the quantitative possibility of state formula by matrix operations based on (1)–(8).

Algorithm 3 is proposed to calculate the cost operators by matrix operations based on (9)–(14).

Now let us analyze the time complexities of our algorithm. We would see the three algorithms as one algorithm and analyze it.

Under the scheduler Adv , we can recursively calculate the truth value of $\|\Phi\|(s)$ in step $\|\Phi\|$, which is the number of the sub-formula of which is recursively defined as below. If $\Phi \in AP \cup \{true\}$, then $|\Phi| = 1$, $|\Phi_1 \wedge \Phi_2| = |\Phi_1| + |\Phi_2| + 1$, $|\neg\Phi| = |\Phi| + 1$, $|\exists \bigcirc \Phi| = |\Phi| + 1$, $|\forall \bigcirc \Phi| = |\Phi| + 1$, $|\exists \Phi_1 \sqcup \Phi_2| = |\Phi_1| + |\Phi_2| + 1$, $|\forall \Phi_1 \sqcup \Phi_2| = |\Phi_1| + |\Phi_2| + 1$, $|\mathbb{E}(=k)| = 1$, $|\mathbb{E}(\leq k)| = k$ and $|\mathbb{E}(\Phi)| = k \times (|\Phi| + 1)$.

The time complexity of calculating the formula $\Phi = a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi$ is only contacted with the size of FDPC M_f and Φ , and is $O(|S|)$. The time of calculating the formula $\Phi = \mathbb{E}(=k) \mid \mathbb{E}(\leq k)$ is only contacted with the size of FDPC M_f and Φ and k , and is $O(|S| \times k)$. The time of calculating the formula $\Phi = \exists \varphi \mid \forall \varphi$ is mainly contacted with the time of calculating the transitive closure of P_{Adv} , e.g., P_{Adv}^* . We use the method of literature [30], and the time complexities is $O(|S|^2 \times \log |S|)$. The time of calculating the formula $\Phi = \mathbb{E}(\Phi)$ is contacted with the time of catch and the time of matrix multiplication, and is $O(|S|^4)$. Above all, we give the time complexities of our algorithm.

Algorithm 3 Calculating the cost operators of FCTL**Require:** a FDPC M_f , step k , a FCTL state formula Φ .**Ensure:** the value of \mathbb{E} .

```

1: if  $\Phi = \mathbb{E}(= k)$  then
2:   Call algorithm 1, put  $s, k, P_{Adv|max|}, P_{Adv|max|}, T_{Adv}$ , get  $t_{k-1}, t_k, T_{Adv}(t_{k-1}, t_k)$ 
3:   return  $(P_{Adv|max|}^k \circ E)(s) \cdot C_{T_{Adv}(t_{k-1}, t_k)}(t_{k-1})$ 
4: end if
5: if  $\Phi = \mathbb{E}(\leq k)$  then
6:   for  $i \Leftarrow 1$  to  $k$  do
7:     Call algorithm 1, put  $s, i, P_{Adv|max|}, P_{Adv|max|}, T_{Adv}$ , get  $t_{i-1}, t_i, T_{Adv}(t_{i-1}, t_i)$ 
8:      $sum \Leftarrow sum + (P_{Adv|max|}^i \circ E)(s) \cdot C_{T_{Adv}(t_{i-1}, t_i)}(t_{i-1})$ 
9:   end for
10:  return  $sum$ 
11: end if
12: if  $\Phi = \mathbb{E}(\Phi)$  then
13:  Call algorithm 1, put  $s, 1, P_{Adv|Sat(\Phi)|max|}, P_{Adv|Sat(\Phi)|max|}, T_{Adv}$ , get  $s_0, s_1, T_{Adv}(s_0, s_1)$ 
14:   $sum(0) \Leftarrow (P_{Adv|Sat(\Phi)|max|} \circ E)(s) \cdot C_{T_{Adv}(s_0, s_1)}(s)$ 
15:  for  $i \Leftarrow 2$  to  $|S| - |Sat(\Phi)|$  do
16:    for  $m \Leftarrow 1$  to  $i - 1$  do
17:      Call algorithm 1, put  $s, m, P'_{Adv|Sat(\Phi)|max|}, P'_{Adv|Sat(\Phi)|max|}, T_{Adv}$ , get
       $s_{m-1}, s_m, T_{Adv}(s_{m-1}, s_m)$ 
18:       $sum(i) \Leftarrow sum(i) + ((P'_{Adv|Sat(\Phi)|max|})^m \circ E)(s) \cdot C_{T_{Adv}(s_{m-1}, s_m)}(s_{m-1})$ 
19:    end for
20:    Call algorithm 1, put  $s, i, P'_{Adv|Sat(\Phi)|max|}, P_{Adv|Sat(\Phi)|max|}, T_{Adv}$ , get
     $s_{i-1}, s_i, T_{Adv}(s_{i-1}, s_i)$ 
21:     $sum(i) \Leftarrow sum(i) + ((P'_{Adv|Sat(\Phi)|max|})^{i-1} \circ P_{Adv|Sat(\Phi)|max|} \circ E)(s) \cdot$ 
     $C_{T_{Adv}(s_{i-1}, s_i)}(s_{i-1})$ 
22:    if  $sum(i) \geq sum(0)$  then
23:       $sum(0) \Leftarrow sum(i)$ 
24:    end if
25:  end for
26:  return  $sum(0)$ 
27: end if

```

Theorem 1. Let $M_f = (S, Act, P, I, AP, L, C)$ be a finite FDPC, Φ be a FCTL formula and k be a natural number. Then, the time complexities of calculating the truth values or expected cost is $O(\text{size}(M_f) \times \text{poly}(S) \times |\Phi| \times k)$, where $\text{size}(M_f)$ is the size of FDPC M_f , $\text{poly}(S)$ is a polynomial of $|S|$, $|\Phi|$ is the number of the sub-formula of $|\Phi|$, and k is the given natural number.

6. Illustrative Examples

A medical expert system is an intelligent computer system that collects, sorts and analyzes a large number of cases by computer, concentrates on the diagnosis results of medical experts, and diagnoses and treats patients. Because of the different judgment standards of each expert for the degree of the patient's conditions and the effect of the treatment plan, using a fuzzy system can reflect the operation process of the system closer to the real world. Figures 10–13 is a simple medical expert system, in which there are three experts. Each expert gives different treatment plans, which are represented by α, β, γ . The model has four states of the patients, respectively, represented by s_0, s_1, s_2, s_3 . The variables in the state indicate the patient's health states, which can be divided into B (bad), G (general), N (normal) and E (enough). Different experts have a different understanding of these four health conditions. Therefore, we give fuzzy values to the four to show the health of patients. When treatment scheme α_i is used in state s_i , cost $C(s_i, \alpha_i)$ will be generated, indicating the

treatment cost of the scheme. When using a single treatment scheme, the state transition of patients is shown in Figures 10–12. When three experts consult, a complex system is synthesized, as shown in Figure 13. The connecting line with the arrow in the figure indicates transition. The transition possibility is given by the number on the connecting line, and the cost is indicated by the underlined number in the figure. For example, $s_0 \xrightarrow{\alpha, 0.8, \underline{160}} s_1$ indicates that the patient is in state s_0 , using treatment scheme α , then the possibility of transition to state s_1 is 0.8, and the treatment cost is 160.

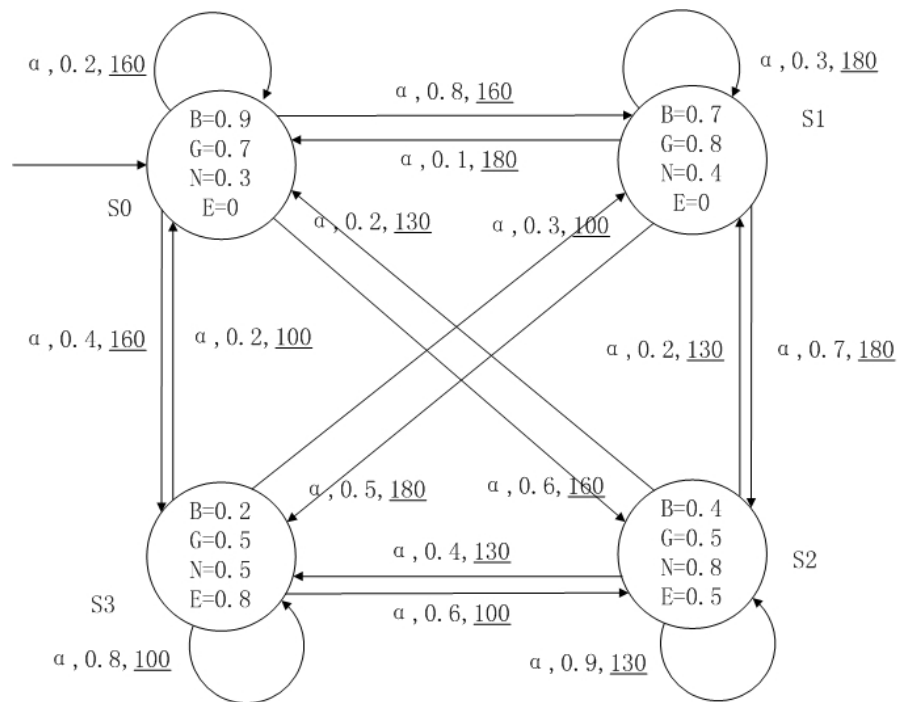


Figure 10. The model of treatment options α .

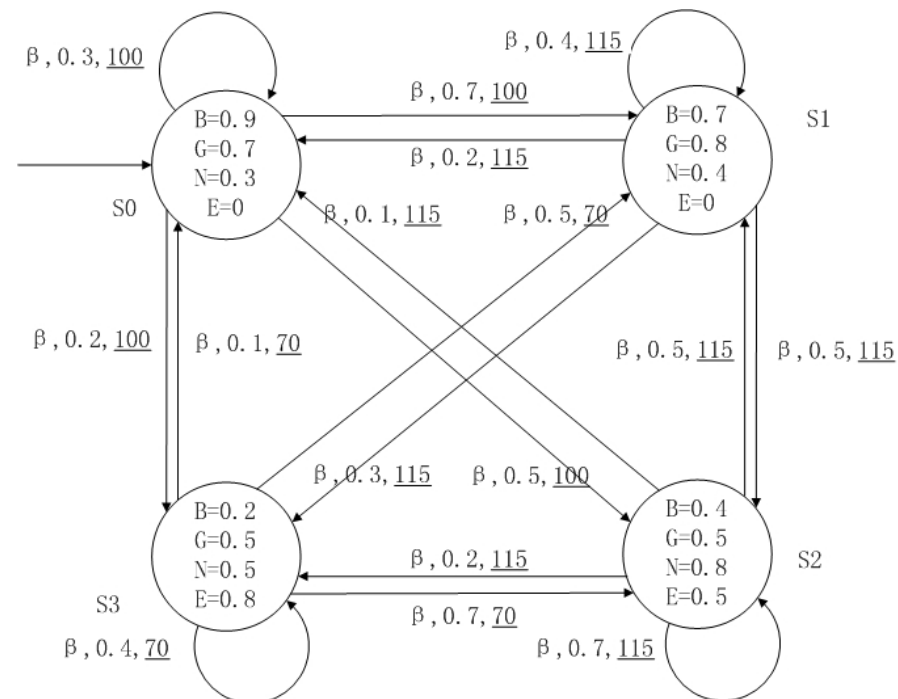


Figure 11. The model of treatment options β .

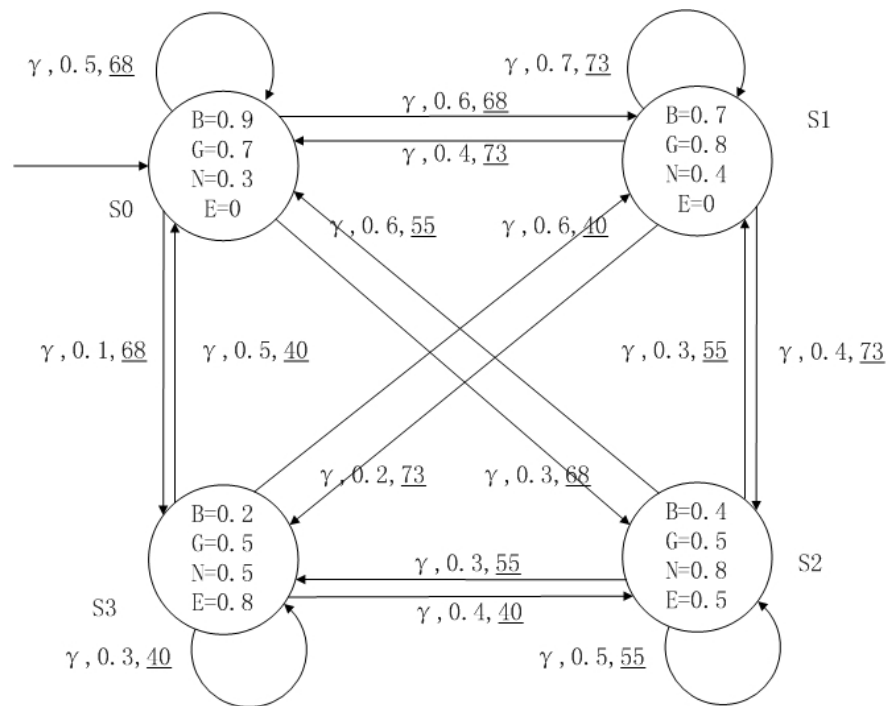


Figure 12. The model of treatment options γ .

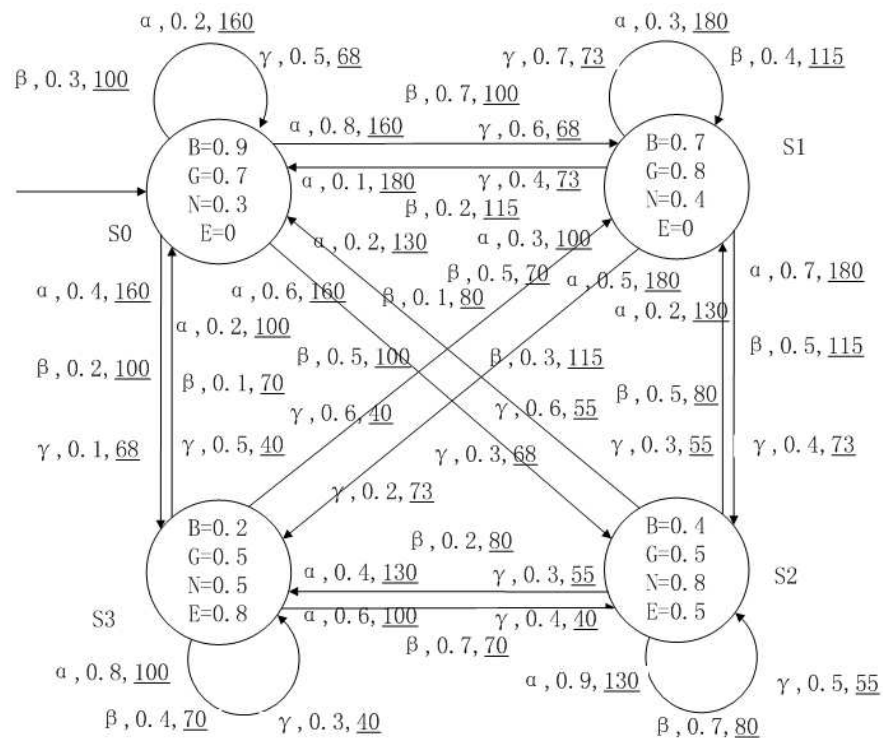


Figure 13. The FDPC of medical expert system.

(1) $\|\exists \circ N\|_{max}(s_2) = 0.8, \|\exists \circ N\|_{min}(s_2) = 0.3$. $\|\exists \circ N\|_{max}(s_2) = 0.8$ is the maximum truth value of that there exists one plan where the patient starts from state s_2 and becomes normal after one treatment. $\|\exists \circ N\|_{min}(s_2) = 0.3$ is the minimum truth value of that there exists one plan where the patient starts from state s_2 and becomes normal after one treatment.

(2) $\|\forall \circ G\|_{max}(s_2) = 0.4, \|\forall \circ G\|_{min}(s_2) = 0.1$. $\|\forall \circ G\|_{max}(s_2) = 0.4$ is the maximum truth value of all of the plans to satisfy that the patient starts from state s_2 and

becomes general after one treatment. $\|\forall \circ G\|_{inx}(s_2) = 0.1$ is the minimum truth value of all of the plans to satisfy that the patient starts from state s_2 and becomes general after one treatment.

(3) $\|\exists G \sqcup E\|_{max}(s_1) = 0.5$, $\|\exists G \sqcup E\|_{min}(s_1) = 0.4$. $\|\exists G \sqcup E\|_{max}(s_1) = 0.5$ is the maximum truth value that there exists one plan that the patient starts from state s_1 , keeps general in treatments and becomes enough finally. $\|\exists G \sqcup E\|_{min}(s_1) = 0.4$ is the minimum truth value that there exists one plan that the patient starts from state s_1 , keeps general in treatments and becomes enough finally.

(4) $\|\forall G \sqcup N\|_{max}(s_1) = 0.4$, $\|\forall G \sqcup N\|_{min}(s_1) = 0.1$. $\|\forall G \sqcup N\|_{max}(s_1) = 0.4$ is the maximum truth value that all of the plans to satisfy that the patient starts from state s_1 , keeps general in treatments and becomes enough finally. $\|\forall G \sqcup N\|_{min}(s_1) = 0.1$ is the minimum truth value that all of the plans to satisfy that the patient starts from state s_1 , keeps general in treatments and becomes enough finally.

(5) $\|\mathbb{E}(= 6)\|_{max}(s_0) = 51.1$, $\|\mathbb{E}(= 6)\|_{min}(s_0) = 22$. $\|\mathbb{E}(= 6)\|_{max}(s_0) = 51.1$ is the maximum skip 6 instantaneous expected cost of that the patient starts from state s_0 . $\|\mathbb{E}(= 6)\|_{min}(s_0) = 22$ is the minimum skip 6 instantaneous expected cost of that the patient starts from state s_0 .

(6) $\|\mathbb{E}(\leq 6)\|_{max}(s_0) = 383.5$, $\|\mathbb{E}(\leq 6)\|_{min}(s_0) = 158$. $\|\mathbb{E}(\leq 6)\|_{max}(s_0) = 383.5$ is the maximum first 6 steps cumulative expected cost of that the patient starts from state s_0 . $\|\mathbb{E}(\leq 6)\|_{min}(s_0) = 158$ is the minimum first 6 steps cumulative expected cost of that the patient starts from state s_0 .

(7) $\|\mathbb{E}(E)\|_{max}(s_0) = 254$, $\|\mathbb{E}(E)\|_{min}(s_0) = 70$. $\|\mathbb{E}(E)\|_{max}(s_0) = 254$ is the maximum cumulative expected cost of that the patient starts from state s_0 and becomes enough finally. $\|\mathbb{E}(E)\|_{min}(s_0) = 70$ is the minimum cumulative expected cost of that the patient starts from state s_0 and becomes enough finally.

7. Conclusions

This paper provides a polynomial model checking algorithm for the verification of some quantitative properties in fuzzy systems in which in any state a nondeterministic choice and cost between fuzzy sets exist. First, we define a fuzzy decision process model with a cost function. This model can describe the cost consumption and other attributes of a fuzzy system. By introducing the definition of the scheduler, we transmit FDPC into a fuzzy Kripke structure. Next, we give the syntax and semantics of fuzzy computation tree logic with a cost operator to describe the properties. Then, using fuzzy matrix and matrix operations, the quantitative calculation of the computation tree logic model checking on the fuzzy decision process model with the cost is introduced, and the corresponding polynomial time algorithm is proposed.

There are several problems that are worth further study. First, it is interesting to consider the linear temporal logic model checking in FDPC. Second, we would like to extend this method used in this paper to multi-objectives model checking. Finally, we will give some case studies on the methods proposed in this paper.

Author Contributions: Conceptualization and Methodology, Z.M. and Z.L.; Formal analysis, W.L., X.L. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: The work is partially supported by National Natural Science Foundation of China (Grant No. 61962001), Graduate Innovation Project of North Minzu University (Grant No. YCX22196 and No. YCX22184), Scientific Research Project of Introducing Personnel from North Minzu University (Grant No.2020KYQD 14), Provincial key R& D of Ningxia science and Technology Department (Grant No. 2019BEB04032), and Provincial Natural Science Foundation of Ningxia (Grant No. 2021A1017).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of (1)–(10)

Proof. (1) $\| \exists \circ \Phi \|_{\max}(s)$
 $= \bigvee_{\pi \in Paths_{\max}(s)} \| \circ \Phi \|(\pi)$
 $= \bigvee_{\pi \in Paths_{\max}(s)} P_{\max}(s, t) \wedge \| \Phi \| (t)$
 $= \bigvee_{t \in S} \left(\bigvee_{\alpha \in Act(s)} P(s, \alpha, t) \wedge \| \Phi \| (t) \right)$
 $= \bigvee_{t \in S} (P_{\alpha_{\max}}(s, t) \wedge \| \Phi \| (t))$
 $= (P_{\alpha_{\max}} \circ P_{\Phi})(s) \quad \square$

Proof. (2) $\| \exists \circ \Phi \|_{\min}(s)$
 $= \bigvee_{\pi \in Paths_{\min}(s)} \| \circ \Phi \|(\pi)$
 $= \bigvee_{\pi \in Paths_{\min}(s)} P_{\min}(s, t) \wedge \| \Phi \| (t)$
 $= \bigvee_{t \in S} \left(\bigwedge_{\alpha \in Act(s)} P(s, \alpha, t) \wedge \| \Phi \| (t) \right)$
 $= \bigvee_{t \in S} (P_{\alpha_{\min}}(s, t) \wedge \| \Phi \| (t))$
 $= (P_{\alpha_{\min}} \circ P_{\Phi})(s) \quad \square$

Proof. (3) $\| \forall \circ \Phi \|_{\max}(s)$
 $= \bigwedge_{\pi \in Paths_{\max}(s)} \| \circ \Phi \|(\pi)$
 $= \bigwedge_{\pi \in Paths_{\max}(s)} P_{\max}(s, t) \wedge \| \Phi \| (t)$
 $= \bigwedge_{t \in S} \left(\bigvee_{\alpha \in Act(s)} P(s, \alpha, t) \wedge \| \Phi \| (t) \right)$
 $= \bigwedge_{t \in S} (P_{\alpha_{\max}}(s, t) \wedge \| \Phi \| (t))$
 $= \left(\begin{array}{c} \bigwedge_{0 \leq k \leq n} (P_{\alpha_{\max}}(s_0, s_k) \wedge \Phi(s_k)) \\ \vdots \\ \bigwedge_{0 \leq k \leq n} (P_{\alpha_{\max}}(s_n, s_k) \wedge \Phi(s_k)) \end{array} \right) (s)$
 $= \left(\begin{array}{c} 1 - \bigvee_{0 \leq k \leq n} (P_{\alpha_{\max}}^c(s_0, s_k) \vee \Phi^c(s_k)) \\ \vdots \\ 1 - \bigvee_{0 \leq k \leq n} (P_{\alpha_{\max}}^c(s_n, s_k) \vee \Phi^c(s_k)) \end{array} \right) (s)$
 $= \left\{ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) - \left(\begin{array}{c} \bigvee_{0 \leq k \leq n} (P_{\alpha_{\max}}^c(s_0, s_k) \vee \Phi^c(s_k)) \\ \vdots \\ \bigvee_{0 \leq k \leq n} (P_{\alpha_{\max}}^c(s_n, s_k) \vee \Phi^c(s_k)) \end{array} \right) \right\} (s)$
 $= \left\{ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) - \left(\begin{array}{ccc} 1 - P_{\alpha_{\max}}(s_0, s_0) \wedge \Phi(s_0) & \cdots & 1 - P_{\alpha_{\max}}(s_0, s_n) \wedge \Phi(s_n) \\ \vdots & \ddots & \vdots \\ 1 - P_{\alpha_{\max}}(s_n, s_0) \wedge \Phi(s_0) & \cdots & 1 - P_{\alpha_{\max}}(s_n, s_n) \wedge \Phi(s_n) \end{array} \right) \circ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) \right\} (s)$
 $= \left\{ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) - \left(\begin{array}{ccc} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{array} \right) - P_{\alpha_{\max}} \circ D_{\Phi} \circ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) \right\} (s)$

$$= \left\{ \left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right) - (P_{\alpha_{\max}} \circ D_{\Phi})^c \circ E \right\} (s)$$

$$= ((P_{\alpha_{\max}} \circ D_{\Phi})^c \circ E)^c (s) \quad \square$$

Proof. (4) $\| \forall \circ \Phi \|_{\min} (s)$

$$= \bigwedge_{\pi \in Paths_{\min}(s)} \| \circ \Phi \| (\pi)$$

$$= \bigwedge_{\pi \in Paths_{\min}(s)} P_{\min}(s, t) \wedge \| \Phi \| (t)$$

$$= \bigwedge_{t \in S} \left(\bigwedge_{\alpha \in Act(s)} P(s, \alpha, t) \wedge \| \Phi \| (t) \right)$$

$$= \bigwedge_{t \in S} (P_{\alpha_{\min}}(s, t) \wedge \| \Phi \| (t))$$

$$= ((P_{\alpha_{\min}} \circ D_{\Phi})^c \circ E)^c (s) \quad \square$$

Proof. (5) $\| \exists \Phi_1 \sqcup \Phi_2 \|_{\max} (s)$

$$= \bigvee_{\pi \in Paths_{\max}(s)} \| \Phi_1 \cup \Phi_2 \| (\pi)$$

$$= \bigvee_{\pi \in Paths_{\max}(s)} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{k<j} P_{\max}(t_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k) \wedge P_{\max}(t_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \bigvee_{t \in S} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{0<k<j} \left(\bigvee_{\alpha_{k-1} \in Act(t_{k-1})} P(t_{k-1}, \alpha_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k) \right) \wedge$$

$$\bigvee_{\alpha_{k-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \bigvee_{t \in S} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{0<k<j} (P_{\alpha_{\max}}(t_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k)) \wedge P_{\alpha_{\max}}(t_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \| \Phi_2 \| (s) \vee \bigvee_{j>0} \bigvee_{t \in S} \left(\bigwedge_{0 \leq k < j} (\| \Phi_1 \| (t_k) \wedge P_{\alpha_{\max}}(t_k, t_{k+1})) \wedge \| \Phi_2 \| (t_j) \right)$$

$$= ((P_{\Phi_2} \cup \bigcup_{j>0} (D_{\Phi_1} \circ P_{\alpha_{\max}})^j \circ P_{\Phi_2})(s))$$

$$= \left(\bigcup_{j \geq 0} (D_{\Phi_1} \circ P_{\alpha_{\max}})^j \circ P_{\Phi_2} \right) (s)$$

$$= ((D_{\Phi_1} \circ P_{\alpha_{\max}})^* \circ P_{\Phi_2})(s) \quad \square$$

Proof. (6) $\| \exists \Phi_1 \sqcup \Phi_2 \|_{\min} (s)$

$$= \bigvee_{\pi \in Paths_{\min}(s)} \| \Phi_1 \cup \Phi_2 \| (\pi)$$

$$= \bigvee_{\pi \in Paths_{\min}(s)} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{k<j} P_{\min}(t_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k) \wedge P_{\min}(t_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \bigvee_{t \in S} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{0<k<j} \left(\bigwedge_{\alpha_{k-1} \in Act(t_{k-1})} P(t_{k-1}, \alpha_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k) \right) \wedge$$

$$\bigwedge_{\alpha_{k-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \bigvee_{t \in S} (\| \Phi_2 \| (s) \vee \bigvee_{j>0} (\| \Phi_1 \| (s) \wedge$$

$$\bigwedge_{0<k<j} (P_{\alpha_{\min}}(t_{k-1}, t_k) \wedge \| \Phi_1 \| (t_k)) \wedge P_{\alpha_{\min}}(t_{j-1}, t_j) \wedge \| \Phi_2 \| (t_j)))$$

$$= \| \Phi_2 \| (s) \vee \bigvee_{j>0} \bigvee_{t \in S} \left(\bigwedge_{0 \leq k < j} (\| \Phi_1 \| (t_k) \wedge P_{\alpha_{\min}}(t_k, t_{k+1})) \wedge \| \Phi_2 \| (t_j) \right)$$

$$= ((P_{\Phi_2} \cup \bigcup_{j>0} (D_{\Phi_1} \circ P_{\alpha_{\min}})^j \circ P_{\Phi_2})(s))$$

$$= \left(\bigcup_{j \geq 0} (D_{\Phi_1} \circ P_{\alpha_{\min}})^j \circ P_{\Phi_2} \right) (s)$$

$$= ((D_{\Phi_1} \circ P_{\alpha_{\min}})^* \circ P_{\Phi_2})(s) \quad \square$$

Proof. $(7) \|\forall \Phi_1 \sqcup \Phi_2\|_{\max}(s)$

$$\begin{aligned}
 &= \bigwedge_{\pi \in Paths_{\max}(s)} \|\Phi_1 \cup \Phi_2\|(\pi) \\
 &= \bigwedge_{\pi \in Paths_{\max}(s)} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \\
 &\quad \bigwedge_{k<j} P_{\max}(t_{k-1}, t_k) \wedge \|\Phi_1\|(t_k) \wedge P_{\max}(t_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
 &= \bigwedge_{t \in S} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \bigwedge_{0<k<j} (\bigvee_{\alpha_{k-1} \in Act(t_{k-1})} P(t_{k-1}, \alpha_{k-1}, t_k) \wedge \\
 &\quad \|\Phi_1\|(t_k) \wedge \bigvee_{\alpha_{k-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
 &= \bigwedge_{t \in S} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \bigwedge_{0<k<j} (P_{\alpha_{\max}}(t_{k-1}, t_k) \wedge \|\Phi_1\|(t_k)) \wedge \\
 &\quad P_{\alpha_{\max}}(t_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
 &= \|\Phi_2\|(s) \vee \bigvee_{j>0} \bigwedge_{t \in S} ((\bigwedge_{0 \leq k < j} (\|\Phi_1\|(t_k) \wedge P_{\alpha_{\max}}(t_k, t_{k+1})) \wedge \|\Phi_2\|(t_j))) \\
 &= \|\Phi_2\|(s) \vee \bigvee_{j>0} \begin{pmatrix} K_0 \\ \vdots \\ K_n \end{pmatrix} (s) \\
 &= \|\Phi_2\|(s) \vee \bigvee_{j>0} \{E - \begin{pmatrix} J_{00} \dots & J_{0n} \\ \vdots & \vdots \\ J_{n0} \dots & J_{nn} \end{pmatrix} (s) \circ E\} (s) \\
 &= \|\Phi_2\|(s) \vee \bigvee_{j>0} \{E - (D_S - \begin{pmatrix} L_0 \dots & L_0 \\ \vdots & \vdots \\ L_n \dots & L_n \end{pmatrix}) \circ \\
 &\quad \begin{pmatrix} O_0 \dots & O_n \\ \vdots & \vdots \\ O_0 \dots & O_n \end{pmatrix}) \circ \dots \circ \begin{pmatrix} O_0 \dots & O_n \\ \vdots & \vdots \\ O_0 \dots & O_n \end{pmatrix} \circ D_{\Phi_2} \circ E\} (s) \\
 &= \|\Phi_2\|(s) \vee \{E - [D_S - \\
 &\quad ((D_{\Phi_1} \circ P_{\alpha_{\max}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{\max}})^c)^c)^* \circ D_{\Phi_2}] \circ E\} (s) \\
 &= \|\Phi_2\|(s) \vee [(((D_{\Phi_1} \circ P_{\alpha_{\max}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{\max}})^c)^c)^* \circ D_{\Phi_2}]^c \\
 &\quad \circ E)^c (s) \\
 &= (P_{\Phi_2} \cup [(((D_{\Phi_1} \circ P_{\alpha_{\max}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{\max}})^c)^c)^* \circ D_{\Phi_2}]^c \\
 &\quad \circ E)^c (s)
 \end{aligned}$$

where,

$$\begin{aligned}
 K_0 &= \bigwedge_{0 \leq u \leq n} (\|\Phi\|(s_0) \wedge \bigwedge_{0 \leq k \leq n} P_{\alpha_{\max}}(s_0, s_k)) \wedge \bigwedge_{0 \leq m \leq n} \\
 &\quad (\|\Phi\|(s_k) \wedge P_{\alpha_{\max}}(s_k, s_m)) \wedge \dots \wedge \bigwedge_{0 \leq o \leq n} (\|\Phi\|(s_o) \wedge \\
 &\quad P_{\alpha_{\max}}(s_o, s_u)) \wedge \|\Phi_2\|(s_u) \\
 K_n &= \bigwedge_{0 \leq u \leq n} (\|\Phi\|(s_n) \wedge \bigwedge_{0 \leq k \leq n} P_{\alpha_{\max}}(s_0, s_k)) \wedge \bigwedge_{0 \leq m \leq n} \\
 &\quad (\|\Phi\|(s_k) \wedge P_{\alpha_{\max}}(s_k, s_m)) \wedge \dots \wedge \bigwedge_{0 \leq o \leq n} (\|\Phi\|(s_o) \wedge \\
 &\quad P_{\alpha_{\max}}(s_o, s_u)) \wedge \|\Phi_2\|(s_u) \\
 J_{00} &= (\|\Phi_1\|^c(s_0) \bigvee_{0 \leq k \leq n} P_{\alpha_{\max}}^c(s_0, s_k)) \bigvee_{0 \leq m \leq n} \\
 &\quad (\|\Phi_1\|^c(s_k) \vee P_{\alpha_{\max}}^c(s_k, s_m)) \vee \dots \vee_{0 \leq o \leq n} (\|\Phi_1\|^c(s_o) \vee P_{\alpha_{\max}}^c(s_o, s_0)) \vee \|\Phi_2\|^c(s_0) \\
 J_{n0} &= (\|\Phi_1\|^c(s_n) \bigvee_{0 \leq k \leq n} P_{\alpha_{\max}}^c(s_n, s_k)) \bigvee_{0 \leq m \leq n} \\
 &\quad (\|\Phi_1\|^c(s_k) \vee P_{\alpha_{\max}}^c(s_k, s_m)) \vee \dots \vee_{0 \leq o \leq n} (\|\Phi_1\|^c(s_o) \vee P_{\alpha_{\max}}^c(s_o, s_0)) \vee \|\Phi_2\|^c(s_0) \\
 J_{0n} &= (\|\Phi_1\|^c(s_0) \bigvee_{0 \leq k \leq n} P_{\alpha_{\max}}^c(s_0, s_k)) \bigvee_{0 \leq m \leq n} \\
 &\quad (\|\Phi_1\|^c(s_k) \vee P_{\alpha_{\max}}^c(s_k, s_m)) \vee \dots \vee_{0 \leq o \leq n} (\|\Phi_1\|^c(s_o) \vee P_{\alpha_{\max}}^c(s_o, s_n)) \vee \|\Phi_2\|^c(s_n) \\
 J_{nn} &= (\|\Phi_1\|^c(s_n) \bigvee_{0 \leq k \leq n} P_{\alpha_{\max}}^c(s_n, s_k)) \bigvee_{0 \leq m \leq n} \\
 &\quad (\|\Phi_1\|^c(s_k) \vee P_{\alpha_{\max}}^c(s_k, s_m)) \vee \dots \vee_{0 \leq o \leq n} (\|\Phi_1\|^c(s_o) \vee P_{\alpha_{\max}}^c(s_o, s_n)) \vee \|\Phi_2\|^c(s_n)
 \end{aligned}$$

$$\begin{aligned}
L_0 &= \|\Phi_1\| (s_0) \bigwedge_{0 \leq k \leq n} P_{\alpha_{\max}}(s_n, s_k) \\
L_n &= \|\Phi_1\| (s_n) \bigwedge_{0 \leq k \leq n} P_{\alpha_{\max}}(s_n, s_k) \\
O_0 &= \bigwedge_{0 \leq k \leq n} \|\Phi_1\| (s_k) \wedge P_{\alpha_{\max}}(s_k, s_0) \\
O_n &= \bigwedge_{0 \leq k \leq n} \|\Phi_1\| (s_k) \wedge P_{\alpha_{\max}}(s_k, s_n) \quad \square
\end{aligned}$$

Proof. (8) $\|\forall \Phi_1 \sqcup \Phi_2\|_{\min}(s)$

$$\begin{aligned}
&= \bigwedge_{\pi \in Paths_{\min}(s)} \|\Phi_1 \cup \Phi_2\|(\pi) \\
&= \bigwedge_{\pi \in Paths_{\min}(s)} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \\
&\bigwedge_{k<j} P_{\min}(t_{k-1}, t_k) \wedge \|\Phi_1\|(t_k) \wedge P_{\min}(t_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
&= \bigwedge_{t \in S} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \bigwedge_{0 < k < j} (\bigwedge_{\alpha_{k-1} \in Act(t_{k-1})} P(t_{k-1}, \alpha_{k-1}, t_k) \wedge \\
&\|\Phi_1\|(t_k) \wedge \bigwedge_{\alpha_{k-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
&= \bigwedge_{t \in S} (\|\Phi_2\|(s) \vee \bigvee_{j>0} (\|\Phi_1\|(s) \wedge \\
&\bigwedge_{0 < k < j} (P_{\alpha_{\min}}(t_{k-1}, t_k) \wedge \|\Phi_1\|(t_k) \wedge P_{\alpha_{\min}}(t_{j-1}, t_j) \wedge \|\Phi_2\|(t_j))) \\
&= \|\Phi_2\|(s) \vee \bigvee_{j>0} \bigwedge_{t \in S} ((\bigwedge_{0 \leq k < j} (\|\Phi_1\|(t_k) \wedge P_{\alpha_{\min}}(t_k, t_{k+1})) \wedge \|\Phi_2\|(t_j))) \\
&= (P_{\Phi_2} \cup (((D_{\Phi_1} \circ P_{\alpha_{\min}})^c \circ D_S)^c \circ ((D_S \circ (D_{\Phi_1} \circ P_{\alpha_{\min}})^c)^c)^* \circ D_{\Phi_2})^c \\
&\circ E)^c(s) \quad \square
\end{aligned}$$

Proof. (9) $\|\mathbb{E}(=k)\|_{\max}(s)$

$$\begin{aligned}
&= Ex_{\max}^s(cost[=k]) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} P_{\max}(t_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} \bigvee_{\alpha_{j-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} P_{\alpha_{\max}}(t_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (P_{\alpha_{\max}|\max}^k \circ E)(s) \cdot C_{T_{\alpha_{\max}}(t_{k-1}, t_k)}(t_{k-1}) \\
&\text{where } t_m = \operatorname{argmax}_{t_m \in S} (P_{\alpha_{\max}|\max}^m(s, t_m)). \quad \square
\end{aligned}$$

Proof. (10) $\|\mathbb{E}(=k)\|_{\min}(s)$

$$\begin{aligned}
&= Ex_{\min}^s(cost[=k]) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} P_{\min}(t_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} \bigwedge_{\alpha_{j-1} \in Act(t_{j-1})} P(t_{j-1}, \alpha_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (\bigwedge_{0 < j \leq k} (\bigvee_{t \in S} P_{\alpha_{\min}}(t_{j-1}, t_j))) \cdot C(t_{k-1}, \alpha_{k-1}) \\
&= (P_{\alpha_{\min}|\max}^k \circ E)(s) \cdot C_{T_{\alpha_{\min}}(t_{k-1}, t_k)}(t_{k-1}) \\
&\text{where } t_m = \operatorname{argmax}_{t_m \in S} (P_{\alpha_{\min}|\max}^m(s, t_m)). \quad \square
\end{aligned}$$

References

- Baier, C.; Katoen, J.P. *Principles of Model Checking*; MIT Press: Cambridge, MA, USA, 2008.
- Xie, D.; Xiong, W.; Bu, L.; Li, X. Deriving unbounded reachability proof of linear hybrid automata during bounded checking procedure. *IEEE Trans. Comput.* **2017**, *66*, 416–430. <http://dx.doi.org/10.1109/TC.2016.2604308>.
- Rawlings, B.C.; Lafortune, S.; Ydstie, B.E. Supervisory control of labeled transition systems subject to multiple reachability requirements via symbolic model checking. *IEEE Trans. Control. Syst. Technol.* **2020**, *28*, 644–652. <http://dx.doi.org/10.1109/TCST.2018.2877621>.
- Hart, S.; Sharir, M. Probabilistic propositional temporal logics. *Inf. Control.* **1986**, *70*, 97–155. [http://dx.doi.org/10.1016/S0019-9958\(86\)80001-8](http://dx.doi.org/10.1016/S0019-9958(86)80001-8).
- Alfaro, L.D.; Kwiatkowska, M.; Norman, G.; Parker, D.; Segala, R. Symbolic model checking of probabilistic processes using mtbddds and the kronecker representation. In *Tools and Algorithms for the Construction and Analysis of Systems*; Graf, S., Schwartzbach, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 395–410. http://dx.doi.org/10.1007/3-540-46419-0_27.

6. Baier, C.; Clarke, E.M.; Hartonas-Garmhausen, V.; Kwiatkowska, M.; Ryan, M., Symbolic model checking for probabilistic processes. In *International Colloquium on Automata, Languages, and Programming*; Degano, P., Gorrieri, R., Marchetti-Spaccamela, A., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; pp. 430–440. http://dx.doi.org/10.1007/3-540-63165-8_199.
7. Baier, C.; Ciesinski, F.; Grosser, M. ProbMela: A Modeling Language for Communicating Probabilistic Processes. In *Proceedings of the Second ACM and IEEE International Conference on Formal Methods and Models for Co-Design*, San Diego, CA, USA, 23–25 June 2004; Volume 4, pp. 57–66. <http://dx.doi.org/10.1109/MEMCOD.2004.1459815>.
8. Baier, C.; de Alfaro, L.; Forejt, V.; Kwiatkowska, M. *Model Checking Probabilistic Systems*, Springer International Publishing: Cham, Switzerland, 2018; pp. 963–999. http://dx.doi.org/10.1007/978-3-319-10575-8_28.
9. Chechik, M.; Gurfinkel, A.; Devereux, B.; χ chek: A Multi-Valued Model-Checker. In *Computer Aided Verification*; Brinksma, E., Larsen, K.G., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 505–509. http://dx.doi.org/10.1007/3-540-45657-0_41.
10. Chechik, M.; Easterbrook, S.; Devereux, B. Model checking with multi-valued temporal logics. In *Proceedings of the 31st IEEE International Symposium on Multiple-Valued Logic*, Warsaw, Poland, 22–24 May 2001; IEEE: Warsaw, Poland, 2001; pp. 187–192. <http://dx.doi.org/10.1109/ISMVL.2001.924571>.
11. Li, Y.; Droste, M.; Lei, L. Model checking of linear-time properties in multi-valued systems. *Inf. Sci.* **2017**, *377*, 51–74. <http://dx.doi.org/10.1016/j.ins.2016.10.030>.
12. Chechik, M.; Gurfinkel, A.; Devereux, B.; Lai, A.; Easterbrook, S. Data structures for symbolic multi-valued model-checking. *Form. Methods Syst. Des.* **2006**, *29*, 295–344. <http://dx.doi.org/10.1007/s10703-006-0016-z>.
13. Li, Y.; Li, L. Model checking of linear-time properties based on possibility measure. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 842–854. <http://dx.doi.org/10.1109/TFUZZ.2012.2232298>.
14. Li, Y.; Li, Y.; Ma, Z. Computation tree logic model checking based on possibility measures. *Fuzzy Sets Syst.* **2015**, *262*, 44–59. <http://dx.doi.org/10.1016/j.fss.2014.03.009>.
15. Li, Y.; Ma, Z. Quantitative computation tree logic model checking based on generalized possibility measures. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 2034–2047. <http://dx.doi.org/10.1109/TFUZZ.2015.2396537>.
16. Zhang, S.; Li, Y. Expressive power of linear-temporal logic based on generalized possibility measures. In *Proceedings of the 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, BC, Canada, 24–29 July 2016; pp. 431–436. <http://dx.doi.org/10.1109/FUZZ-IEEE.2016.7737719>.
17. Subhankar Mukherjee, P.D. A fuzzy real-time temporal logic. *Int. J. Approx. Reason.* **2013**, *54*, 1452–1470. <http://dx.doi.org/10.1016/j.ijar.2013.06.004>.
18. Yang, C.; Li, Y. ϵ -bisimulation relations for fuzzy automata. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 2017–2029. <http://dx.doi.org/10.1109/TFUZZ.2017.2760278>.
19. Pan, H.; Li, Y.; Cao, Y.; Ma, Z. Model checking fuzzy computation tree logic. *Fuzzy Sets Syst.* **2015**, *262*, 60–77. <http://dx.doi.org/10.1016/j.fss.2014.07.008>.
20. Zadeh, L. Fuzzy sets. *Inf. Cont.* **1965**, *8*, 338–353. [http://dx.doi.org/10.1016/S0019-9958\(65\)90241-X](http://dx.doi.org/10.1016/S0019-9958(65)90241-X).
21. Zimmermann, H.-J. *Fuzzy Set Theory and Its Applications*; Springer: Dordrecht, The Netherlands, 2001. <http://dx.doi.org/10.1007/978-94-010-0646-0>.
22. Frigeri, A.; Pasquale, L.; Spoletini, P. Fuzzy time in linear temporal logic. *ACM Trans. Comput. Log.* **2014**, *15*, 1–22. <http://dx.doi.org/10.1145/2629606>.
23. Garmendia Salvador, L.; González del Campo, R.; López, V.; Recasens Ferrés, J. An algorithm to compute the transitive closure, a transitive approximation and a transitive opening of a proximity. *Mathw. Soft Comput.* **2009**, *16*, 175–191.
24. 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 Intl Conference on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, Fukuoka, Japan, 5–8 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 449–453. <http://dx.doi.org/10.1109/DASC/PiCom/CBDCCom/CyberSciTech.2019.00091>.
25. Choi, J.; Lee, K.Y. *Fuzzy Decision Making for TEP*; Wiley-IEEE Press: Piscataway, NJ, USA, 2022; pp. 375–399. <http://dx.doi.org/10.1002/9781119819042.ch15>.
26. Li, Y.; Wei, J. Possibilistic fuzzy linear temporal logic and its model checking. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1899–1913. <http://dx.doi.org/10.1109/TFUZZ.2020.2988848>.
27. Yang, W.; Jhang, S.; Shi, S.E.A. A novel additive consistency for intuitionistic fuzzy preference relations in group decision making. *Appl. Intell.* **2020**, *50*, 4342–4356. <http://dx.doi.org/10.1007/s10489-020-01796-z3>.
28. Xue, Y.; Deng, Y. Decision making under measure-based granular uncertainty with intuitionistic fuzzy sets. *Appl. Intell.* **2021**, *51*, 6224–6233. <http://dx.doi.org/10.1007/s10489-021-02216-6>.
29. Li, Y.M. *Analysis of Fuzzy Systems*; Science Press: Beijing, China, 2005. (In Chinese)
30. Zhang, Y.; Tang, J.; Meng, F. Programming model-based method for ranking objects from group decision making with interval-valued hesitant fuzzy preference relations. *Appl. Intell.* **2019**, *49*, 837–857. <http://dx.doi.org/10.1007/s10489-018-1292-1>.