Article

Research on Fuzzy Computation Tree Temporal Logic with Quality Constraints and Its Model Checking

XianFeng Yu , YongMing Li [*] , ShengLing Geng , HuiRong Li

Posted Date: 24 October 2024

doi: 10.20944/preprints202410.1921.v1

Keywords: Quality function; Quality predicate; QFCTL; QFCTL*; Model checking

Article

Research on Fuzzy Computation Tree Temporal Logic with Quality Constraints and Its Model Checking

XianFeng Yu [1,2], YongMing Li [1,3,*], ShengLing Geng [1] and HuiRong Li [2]

[1] Sollege of Computer Science, Qinghai Normal University, Xining 810008, China
[2] School of Mathematics and Computer Application, Shangluo University, Shangluo, 726000, China
[3] School of Mathematics and Statistics, Shaanxi Normal University, Xian 710062, China
* Correspondence: pioneer.369@163.com

Abstract: The encapsulation of particular quality functions and predicates within temporal logic formulas markedly enhances the representation of detailed temporal characteristics within a system. During our preliminary investigations, we innovatively combined quality constraint functions and predicates with Possibility Linear Temporal Logic (PoLTL), yielding the conception of Fuzzy Linear Temporal Logic with Quality Constraints (QFLTL). This amalgamation results in a significant elevation of QFLTL's expressivity relative to PoLTL, ensuring the preservation of informational integrity whilst achieving a synchronized, yet selectively inclined, and exact consolidation of path reachability specifics alongside property satisfaction evaluations. This treatise represents a significant contribution to the field by integrating quality constraint functions and predicates into Possibility Computation Tree Temporal Logic (PoCTL), thus giving rise to Fuzzy Computation Tree Temporal Logic with Quality Constraints (QFCTL). We provide a comprehensive definition of QFCTL's syntax, conduct an in-depth analysis of its logical characteristics, outline a precise model checking algorithm for QFCTL, and perform a meticulous complexity assessment of said algorithm. Moreover, by enriching QFCTL with a quantitative characteristic predicate operator, we innovate, culminating in the development of an enhanced Fuzzy Computation Tree Temporal Logic with Quality Constraints (QFCTL*). The treatise explores the logical facets of QFCTL*, formulates a bespoke model checking algorithm for QFCTL*, and conducts a rigorous analysis of its algorithmic complexity. To attest to the practical utility and robust expressive power of QFCTL and QFCTL*, we present a model checking example that serves as empirical evidence of the efficacy of the proposed model checking algorithms.

Keywords: quality function; quality predicate; QFCTL; QFCTL*; model checking

MSC: 68T37 Reasoning under uncertainty in the context of artificial intelligence

1. Introduction

Model checking [1,2] is a cornerstone in formal verification, prominently featuring model checking for Linear Temporal Logic (LTL) and Computation Tree Logic (CTL). The automation inherent in model checking has facilitated its widespread adoption in the analysis and validation of software and hardware systems[3–5], communication protocols[6,7], and security protocols[8,9], achieving commendable outcomes. As computer systems grow in scale and complexity, practical systems increasingly exhibit quantitative behavioral attributes. Multi-agent systems [10–13], for instance, feature intricate dynamic structures and behavioral patterns that require the integration of quantitative information for an accurate portrayal of their dynamic behavior. In addressing the verification challenges posed by systems endowed with quantitative data, quantitative model checking methodologies have piqued considerable interest from academic and industrial circles alike. Hart et al. [14,15] pioneered probabilistic model checking, rooted in probability measures, employing Markov chains or Markov decision processes to model system behaviors and probabilistic computation tree logic or probabilistic linear temporal logic to define system properties. Sultan et al. [16,17] extended the probabilistic framework to multi-agent systems, introducing probabilistic multi-agent model checking. Chechik et al. [18,19] explored model checking for CTL and LTL on multi-valued Kripke structures, where values reside within a finite De Morgan algebra. Moreover, Li

YongMing et al. established the theoretical foundation for possibility temporal logic and its applications in model checking. Literature [20] proposed Possibility Computation Tree Temporal Logic (PoCTL) and investigated its expressiveness. It was proven that PoCTL, particularly qualitative PoCTL, is more expressive than CTL. Equivalences for expressing fundamental CTL formulas using qualitative PoCTL formulas were provided. Some PoCTL formulas that cannot be expressed by any CTL formula were given. Qualitative properties of repeat accessibility and persistence were represented with PoCTL formulas. A model-checking approach for PoCTL based on fuzzy matrix composite operations and fixed-point techniques was presented, and the time complexity of the algorithm was analyzed. Literature [21] introduced the theory of generalized possibility measures and Generalized Possibility Kripke Structure (GPKS). Generalized Possibility Computation Tree Temporal Logic(GPoCTL) was proposed, with its syntax and semantics defined. A model-checking method for GPoCTL based on fuzzy matrix composite operations and fixed-point techniques was provided, and the time complexity of the algorithm was analyzed.

Quality constraints serve to delineate specific quality requisites that systems or computation trees must satisfy [22,23]. These requisites encompass a spectrum of performance metrics—such as response time and throughput—reliability metrics—such as failure rates—and security metrics. In our pioneering research pursuits, we innovatively amalgamated specific quality functions and predicates into Possibility Linear Temporal Logic (PoLTL), culminating in the proposition of Fuzzy Linear Temporal Logic with Quality Constraints (QFLTL) [24]. QFLTL empowers the articulation of a system's more intricate temporal attributes. These functions and predicates are meticulously defined based on the system's empirical operational data or projected objectives, yielding fuzzy truth values indicative of quality levels. The introduction of quality constraints significantly bolsters QFLTL's expressive capabilities in several facets: (1) Departing from PoLTL's singular information amalgamation operator, the minimum "$\wedge$", QFLTL introduces an enhanced repertoire of information aggregation operators, encompassing minimum "$\wedge$", product "$\cdot$", and the weighted average operator "$\oplus_\lambda$". This innovation effectively mitigates the issue of information erosion. (2) The weighted average operator "$\oplus_\lambda$" enables a preference-sensitive integration of path reachability insights and property satisfaction measures or the synthesis of disparate property subformulas, facilitating the delineation of more nuanced temporal properties. (3) The linear differential operators $\lambda_{cp}(\cdot), \lambda_{ne}(\cdot)$ and $\lambda_{cf}(\cdot)$, coupled with the quality predicate " P ", can impose quality constraints on property formulas grounded in anticipated objectives, thereby enriching the portrayal of diverse system quantifiable properties. (4) By recursively defining formula satisfaction values onto path segments that meet property conditions, it ensures the synchronous alignment of path reachability and property satisfaction. These enhancements have been empirically validated in reference [24].

Building on the robust foundation established by QFLTL research, we introduce quality functions and quality predicates into PoCTL, culminating in the proposition of QFCTL. We delve into its logical characteristics and investigate the model checking theory of QFCTL on Fuzzy Kripke Structure(FKS). Section 2 delineates the syntax and semantics of QFCTL, illustrating its robust expressive capabilities and practical utility through instructive examples. Section 3 explores the intricate logical properties of QFCTL, encompassing the equivalence calculation and partial order relations of its formulas, the functional completeness of QFCTL operators, and the fundamental properties and determination of the characteristic predicate operator $Q_P(\cdot)$, which delineates the fulfillment characteristics of property formulas under the constraint of quality predicate P on pertinent paths, indicating full, none, or partial satisfaction. Section 4 focuses on the model checking problem of QFCTL on FKS. By ingeniously reducing the computation of QFCTL formulas from infinite to finite paths, we introduce a model checking algorithm with a time complexity proportional to the product of the QFCTL formula's length and the FKS's scale, and a space complexity equivalent to the FKS's scale. The reasonableness and complexity of the algorithm are rigorously substantiated. Section 5 builds upon QFCTL by introducing the quantitative characteristic predicate operator $Q_P^*(.)$, leading to the proposal of QFCTL*. We investigate the logical properties of QFCTL*, provide a determination theorem for the operator $Q_P^*(.)$, which describes the fulfillment status of property formula ψ under the constraint of quality predicate P on pertinent paths, signifying the ratio of paths satisfying ψ under P constraint among all paths satisfying ψ. We prove the well-defined nature of $Q_P^*(.)$, delineate the model checking algorithm for QFCTL*, and conduct a thorough analysis of the algorithm's complexity. Section 6 presents model checking application examples of ten illustrative QFCTL formulas and QFCTL* formulas on the FKS that characterizes the patient treatment process, as outlined in reference [24]. Through these applications, we vividly demonstrate the robust expressive capabilities and practical applicability of QFCTL and QFCTL*. Simultaneously, the automatic and effective nature of the model checking algorithms for QFCTL and QFCTL* presented in the paper is empirically validated. Conclusively, in Section 7, we summarize the primary research content of the article and outline some meaningful research directions pertaining to temporal logic with quality constraints and its reasoning problems.

2. The Syntax and Semantics of QFCTL

We introduce these quality functions and quality predicates into PoCTL to propose QFCTL, and present the syntax and semantics of QFCTL.

2.1. The Syntax of QFCTL

In a concerted effort to enhance the paper's reachability and readability, we will first revisit the fuzzy propositional operators delineated in reference [24].

Definition1(Fuzzy Operations [24]) $\forall x, y, \lambda \in [0,1]$, *the fuzzy propositional operators are defined as follows:*

(1) $\lambda_{cp}(x) = \lambda \cdot x$;
(2) $\lambda_{ne}(x) = \lambda \cdot x + 1 - \lambda$;
(3) $\lambda_{cf}(x) = \lambda \cdot x + (1 - \lambda)/2$;
(4) $\neg x = 1 - x$;
(5) $x \wedge y = min\,\{x, y\}$;
(6) $x \vee y = max\,\{x, y\}$;
(7) $x \longrightarrow y = max\,\{1 - x, y\}$
(8) $x \oplus_{\lambda} y = \lambda \cdot x + (1 - \lambda) \cdot y$.

Adhering to the syntactical blueprints established by CTL, FCTL, and PoCTL, QFCTL's formulaic structure should incorporate basic propositions, logical conjunctions (such as AND, OR, NOT), path quantifiers (including EXISTS, FOR ALL), and a suite of temporal operators (comprising NEXT, GLOBALLY, EVENTUALLY, UNTIL, and the like). Building upon this robust framework, we incorporate the fuzzy propositional operators meticulously explicated in reference [24]. We augment this foundation with the introduction of the characteristic predicate operator $Q_P(\cdot)$, a critical quality constraint operator. This operator intricately delineates the fulfillment characteristics of property formulas under the constraint of quality predicate P on pertinent paths, signifying scenarios encompassing full satisfaction, none satisfaction, or partial satisfaction, thereby significantly amplifying QFCTL's expressive capacity.

The syntactic form of QFCTL is defined as follows.

Definition 2 [Syntax of QFCTL] Let AP denote a set of atomic propositions, and $\mathbb{Q}$ represent the set of rational numbers. *$APR \subseteq [0,1] \cap \mathbb{Q}$ signifies a finite set of fuzzy propositional constants. Univariate fuzzy propositional logic operators are symbolized by $\triangle_1 \in \{\neg, \lambda_{cp}, \lambda_{ne}, \lambda_{cf}\}$, and bivariate fuzzy propositional logic operators by $\triangle_2 \in \{\wedge, \vee, \longrightarrow, \oplus_{\lambda}\}$. Temporal logic operators are designated by $\Delta \in \{\bigcirc, \Diamond, \Box\}$. The universal path quantifier is denoted by $\forall$ preceding a path formula, and the existential path quantifier by $\exists$ preceding a path formula. Quality predicates are represented by $P \subseteq [0,1] \cap \mathbb{Q}$, and $Q_P(\cdot)$ signifies the characteristic predicate operator.*

QFCTL state formulas φ are recursively defined over $AP \cup APR$ as follows:

(1) *An atomic proposition p, where $p \in AP$, is a QFCTL state formula.*
(2) *A propositional constant r, where $r \in APR$, is a QFCTL state formula, with special cases for $r = 0$ corresponding to "False" and $r = 1$ to "True".*
(3) *Formulas $\triangle_1\, \varphi, \varphi_1 \triangle_2\, \varphi_2$are QFCTL state formulas, where $\varphi, \varphi_1, \varphi_2$ are QFCTL state formulas.*
(4) *Expressions $\forall\psi, \exists\psi, Q_P(\psi)$ are QFCTL state formulas, where ψ is a QFCTL path formula.*

QFCTL path formulas ψ are recursively defined as follows:

(5) *A QFCTL state formula φ is a path formula.*

(6) *Formulas* $\Delta\varphi, \varphi_1 \sqcup \varphi_2$ *are QFCTL path formulas, where* $\varphi, \varphi_1, \varphi_2$ *are QFCTL state formulas.*

To facilitate discussions regarding the complexity of the QFCTL model checking algorithm, the concept of a QFCTL formula's length is defined as follows.

Definition 3 [Length of a QFCTL Formula] *Let* φ *be a QFCTL state formula. The length of* φ*, denoted as* $|\varphi|$*, is defined recursively as follows,*

(1) $|p| = |r| = 1;$
(2) $|\triangle_1 \varphi| = |\varphi| + 1;$
(3) $|\varphi_1 \triangle_2 \varphi_2| = |\varphi_1| + |\varphi_2| + 1;$
(4) $|\forall \triangle_1 \varphi| = |\exists \triangle_1 \varphi| = |Q_P(\triangle_1 \varphi)| = |\forall\Delta\varphi| = |\exists\Delta\varphi| = |Q_P(\Delta\varphi)| = |\varphi| + 1;$
(5) $|\forall(\varphi_1 \triangle_2 \varphi_2)| = |\exists(\varphi_1 \triangle_2 \varphi_2)| = |Q_P(\varphi_1 \triangle_2 \varphi_2)|;$
$= |\forall(\varphi_1 \sqcup \varphi_2)| = |\exists(\varphi_1 \sqcup \varphi_2)| = |Q_P(\varphi_1 \sqcup \varphi_2)| = |\varphi_1| + |\varphi_2| + 1.$

Note 1: *The focus on state formulas when defining the length of QFCTL formulas is due to the model checking process which assesses the satisfaction value of formulas in specific states. Path formulas are integral components, nested within state formulas. According to Definition 2, part (4), path formulas* ψ *are structured as state formulas under the constraints imposed by quantifiers and characteristic predicates.*

Example 1. *Given an atomic proposition set* $AP = \{request, grant\}$*, consider the QFCTL formula below,*

$$\varphi = \forall\Box(request \longrightarrow \exists\Diamond(grant \oplus_{3/4} \bigcirc grant) \wedge \neg(4/5)_{cp}(\forall\Box\neg request))$$

The process for computing the length of formula φ is as follows,

$|\varphi| = (|request \longrightarrow \exists\Diamond(grant \oplus_{3/4} \bigcirc grant)| + 1) + (|\forall\Box\neg request| + 2) + 1$

$= |request \longrightarrow \exists\Diamond(grant \oplus_{3/4} \bigcirc grant)| + |\forall\Box\neg request| + 4$

$= (1 + |\exists\Diamond(grant \oplus_{3/4} \bigcirc grant)| + 1) + (1 + |\neg request|) + 4$

$= |\exists\Diamond(grant \oplus_{3/4} \bigcirc grant)| + |\neg request| + 7$

$= (1 + |grant \oplus_{3/4} \bigcirc grant|) + 2 + 7$

$= |grant \oplus_{3/4} \bigcirc grant| + 10$

$= 1 + |\bigcirc grant| + 1 + 10$

$= |\bigcirc grant| + 12$

$= 14.$

2.2. The Semantics of QFCTL

The semantics of QFCTL is anchored in fuzzy Kripke structures, wherein propositions are mapped onto states via a fuzzy valuation function to determine fuzzy truth values. For a QFCTL formula, its truth value at a particular state is determined by recursively computing the truth values of its constituent subformulas. Let's begin by revisiting the definition of a Fuzzy Kripke Structure (FKS), essential for grounding our understanding of QFCTL's semantic framework.

Definition 4 [Fuzzy Kripke Structures (FKSs)][24]. *An FKS is a tuple* $M = (S, I, \delta, AP, L)$*, where,*

(1) S *is a finite set of states;*

(2) *The fuzzy distribution* $I: S \longrightarrow [0,1]$ *represents the fuzzy set of each state as the initial state;*

(3) $\delta: S \times S \longrightarrow [0,1]$ *represents the fuzzy transition relationship between system states;*

(4) AP *is a set of finite atomic propositions;*

(5) $L: S \longrightarrow [0,1]^{AP}$ *is a state label function that characterizes a set of fuzzy atomic propositions.*

Definition 5 [path and path reachability]. *Suppose* $M = (S, I, \delta, AP, L)$ *is an FKS, where a path* π *is a state sequence* $\pi = \pi_0, \pi_1, \cdots, \pi_i, \pi_{i+1}, \cdots \in S^\omega$ *, and* $L(\pi) \in [0,1]^{AP}$ *represents a set of fuzzy atomic propositions as a fuzzy label function.* $\forall p \in AP, i \in \mathbb{N}$ *(where* $\mathbb{N}$ *represents the set of natural numbers)* $L(\pi_i)(p) \in [0,1]$ *represents the fuzzy atomic proposition induced by atomic proposition* p *on state* π_i*.*

$\pi^i = \pi_i, \pi_{i+1}, \cdots \in S^\omega$ represents a path starting from state π_i. The recursive definition of path reachability is as follows.

$$\delta_i^*(\pi) = \begin{cases} I(\pi_0) & i = 0; \\ \delta_{i-1}^*(\pi) \wedge \delta(\pi_{i-1}, \pi_i) & i > 0. \end{cases}$$

where $\delta_i^*(\pi)$ represents the reachability of the path fragment $\pi_0, \pi_1, \cdots, \pi_i$. This reflects the idea of the "barrel principle", which states that the overall reachability of a path is determined by the minimum reachability of any path fragment. When $i \longrightarrow +\infty$, $\delta_\infty^*(\pi)$ represents the reachability of the infinite path $\pi = \pi_0, \pi_1, \cdots, \pi_i, \pi_{i+1}, \cdots \in S^\omega$.

$Path(M) = \{\pi | \pi \in S^\omega, I(\pi_0) > 0\}$ is the set of infinite paths in M. For every state s within S, $Path(s) = \{\pi | \pi \in S^\omega, s = \pi_0\}$ signifies the collection of all infinite paths π starting at s, whereas $Child(s) = \{s' | \delta(s, s') > 0\}$ designates the ensemble of states s' that are immediately accessible from state s, as defined by the transition function δ.

For every i belonging to the set of natural numbers $\mathbb{N}$, determine the fulfillment value of the QFCTL state formula φ over the path segment denoted by $\pi^i = \pi_i, \pi_{i+1}, \cdots \in S^\omega$ within the infinite sequence of states S^ω. This involves synthesizing the path reachability $\delta_i^*(\pi)$ of the path from the initial state π_0 up to the current state π_i with the fulfillment value of the formula φ at state π_i.

Definition 6. [Semantics for QFCTL] *Let $M = (S, I, \delta, AP, L)$ be a Fuzzy Kripke Structure (FKS), where $P \subseteq [0,1] \cap \mathbb{Q}$ signifies a quality predicate. Let φ represent a QFCTL state formula and ψ a QFCTL path formula. The semantics of QFCTL can be characterized by a fuzzy function: $[\![M, \varphi]\!]: S \rightarrow [0,1]$. Employing an information fusion operator $\circ \in \{\wedge, \cdot, \oplus_\lambda\}$, facilitates the combination of fuzzy truth values. The interpretation $[\![\cdot]\!]$ is recursively defined as follows,*

(1) $[\![s, r]\!] = r$;
(2) $[\![s, p]\!] = L(s)(p)$;
(3) $[\![s, \triangle_1 (\varphi)]\!] = \triangle_1 ([\![s, \varphi]\!])$;
(4) $[\![s, \varphi_1 \triangle_2 \varphi_2]\!] = [\![s, \varphi_1]\!] \triangle_2 [\![s, \varphi_2]\!]$;
(5) $[\![\pi^i, \varphi]\!] = \delta_i^*(\pi) \circ [\![\pi_i, \varphi]\!]$;
(6) $[\![\pi^i, \bigcirc\varphi)]\!] = [\![\pi^{i+1}, \varphi]\!]$;
(7) $[\![\pi^i, \Diamond\varphi)]\!] = \bigvee_{j \geq i} [\![\pi^j, \varphi]\!]$;
(8) $[\![\pi^i, \Box\varphi)]\!] = \bigwedge_{j \geq i} [\![\pi^j, \varphi]\!]$;
(9) $[\![\pi^i, \varphi_1 \sqcup \varphi_2]\!] = \bigvee_{j \geq i} \left([\![\pi^j, \varphi_2]\!] \wedge \bigwedge_{i \leq k < j} [\![\pi^k, \varphi_1]\!] \right)$;
(10) $[\![s, \forall\psi]\!] = \bigwedge_{\pi \in Path(s)} [\![\pi, \psi]\!]$;
(11) $[\![s, \exists\psi]\!] = \bigvee_{\pi \in Path(s)} [\![\pi, \psi]\!]$;
(12) $[\![s, Q_P(\psi)]\!] = \begin{cases} 1 & \forall\pi \in Path(s), [\![\pi, \psi]\!] \in P; \\ 0.5 & \exists\pi, \pi' \in Path(s), [\![\pi, \psi]\!] \in P \wedge [\![\pi', \psi]\!] \in P; \\ 0 & \forall\pi \in Path(s), [\![\pi, \psi]\!] \notin P. \end{cases}$

Note 2: (1) I*f the quality constraint operators $\lambda_{cp}(\cdot), \lambda_{ne}(\cdot)$ and $\lambda_{cf}(\cdot)$, the quality property predicate operator $Q_P(\cdot)$, are removed from QFCTL, and the information fusion operator "$\circ$" is restricted to "$\wedge$" only; then QFCTL degrades to PoCTL. It is evident that the set of PoCTL formulas constitutes a true subset of the set of QFCTL formulas.*

(2) *The semantics of QFCTL state formulas, excluding those that contain the quality property predicate operator $Q_P(\cdot)$, are directly defined on the current state without considering path reachability. Essentially, they are interpreted as fuzzy propositional logic.*

(3) *The semantics for QFCTL formulas are recursively defined over paths, with the fifth clause of Definition 6 serving as a pivotal base case in the recursive definition. Formally, it appears as if the path reachability information $\delta_i^*(\pi)$ is synthesized onto the path segment π^i only when the QFCTL path formula*

is distilled to its core—a state formula φ. This information does not seem to feature in other recursive cases. However, the innermost component of any QFCTL formula is invariably a state formula. Consequently, irrespective of the superficial form of the QFCTL formula, the path reachability information will inevitably be synthesized through the process of recursive iteration. To superimpose $\delta_i^*(\pi)$ at each level of the formula would introduce redundancy in reachability information and lead to informational asynchrony, thus compromising the integrity of the recursive structure.

As an example, let us examine the computation of semantics for the "$\sqcup$" operator,

$$[\![\pi^i,\varphi_1 \sqcup \varphi_2]\!] = \bigvee_{j\geq i}\left([\![\pi^j,\varphi_2]\!] \wedge \bigwedge_{i\leq k<j}[\![\pi^k,\varphi_1]\!]\right) = \bigvee_{j\geq i}\left(\delta_j^*(\pi)\circ[\![\pi_j,\varphi_2]\!] \wedge \bigwedge_{i\leq k<j}\delta_k^*(\pi)\circ[\![\pi_k,\varphi_1]\!]\right).$$

The information fusion operator "·" encapsulates the essence of the multiplicative principle. In the quest for the truth value of the QFLTL formula φ on the path $\pi^i = \pi_i, \pi_{i+1}, \cdots \in S^\omega$, one initiates the process by computing the path reachability $\delta_i^*(\pi)$ of the prefix path, spanning from π_0 to ,π_i. Subsequently, this value is multiplied by the satisfaction value $[\![\pi_i,\varphi]\!]$ of the formula at π^i, ensuring a methodical integration of path reachability and formula satisfaction. The information fusion operator "$\oplus_\lambda$", on the other hand, embodies the philosophy of weighted averages. It calls for the sophisticated amalgamation of path reachability and formula satisfaction values, calibrated according to distinct weights, thereby facilitating a nuanced understanding of formula satisfaction across the path.

2.3. Exemplary Illustrations of QFCTL's Practicality

QFCTL formulas embody a rich expressive capacity. Reflect on the QFCTL formula delineated in Example 1:

$$\varphi = \forall\Box(request \longrightarrow \exists\Diamond(grant \oplus_{3/4} \bigcirc grant) \wedge \neg(3/4)_{cp}(\forall\Box\neg request)).$$

Upon confining the atomic propositions '$request$' and '$grant$' to Boolean values of 0 or 1, φ elucidates the process's unvarying adherence to the stipulation that, post-request initiation ("$request$"), there is a temporal sequence in the future where permission is conferred ("$grant$ "). Should permission be granted consecutively twice, the satisfaction value materializes as 1; if granted solely once, the satisfaction value is assuredly not less than 1/4. Conversely, should the process abstain from initiating a request, the satisfaction value is no less than 1/5.

When the atomic propositions '$request$' and '$grant$' assume values over a more extensive domain, and with due consideration paid to varying degrees of path reachability, φ furnishes an array of values. These disparate values encapsulate distinct process execution trajectories, manifestly evidencing that QFCTL wields a more copious expressive potential than PoCTL.

Example 2. *Figure 1 illustrates the Fuzzy Kripke Structure (FKS), as delineated in reference [24], which characterizes the evolution of a patient's health condition throughout treatment. The atomic propositions* $'poor'$*,* $'fine'$*,* $'excenllent'$ *embody the patient's health status, with each state assigning a fuzzy value within the specified interval [0,1] to these propositions, thereby representing a fuzzy proposition. For example, the assignment* $s_1(e) = 0.5$*, signifies that when the patient is in the state* s_1*, their health status has a 0.5 possibility of being "excellent". The orientation of the directed edges delineates the direction of state transition, and the fuzzy values inscribed upon these edges denote the possibilities of transition between interrelated states. Notably, the sole initial state is identified as* s_0*.*

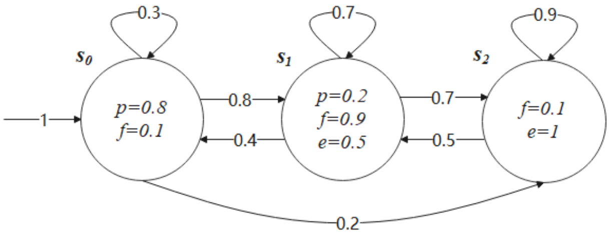

Figure 1. The FKS M for the patient's treatment process.

The subsequent QFCTL formulas are presented, accompanied by their detailed semantic interpretations:

$\varphi_1 = \exists\Diamond(0.8_{cf}(e))$ delineates the maximal possibility that, among all therapeutic alternatives in the medical treatment course, the patient's ultimate health status will be "excellent" with a possibility of not less than 0.8.

$\varphi_2 = \forall\Diamond(0.8_{cf}(e))$ signifies the least possibility that, among all therapeutic alternatives in the medical treatment course, the patient's ultimate health status will be "excellent" with a possibility of not less than 0.8.

$\varphi_3 = \exists\square(f\oplus_{0.4}e)$ encapsulates the highest possibility that, within all therapeutic alternatives, the patient's health status will consistently satisfies the "fine" condition with a 40% weightage, simultaneously alongside the "excellent" condition with a 60% weightage.

$\varphi_4 = \forall\square(f\oplus_{0.4}e)$ illustrates the least possibility that, within all therapeutic alternatives, the patient's health status will persistently adhere to the requirement of being "fine" with a 40% weightage and "excellent" with a 60% weightage.

$\varphi_5 = \exists(0.5_{ne}(f)\sqcup e)$ delineates the maximal possibility that the patient's health status will remain "fine" with a necessity not exceeding 0.5 until it transitions to "excellent."

$\varphi_6 = \forall(0.5_{ne}(f)\sqcup e)$ signifies the least possibility that the patient's health status will remain "fine" with a necessity not exceeding 0.5 until it transitions to "excellent."

$\varphi_7 = Q_{\geq 0.3}(0.8_{cf}(e))$ delineates the fulfillment condition (whether entirely, partially, or not at all) for the possibility of the patient's health status being "excellent" with a possibility of not more than 0.8, with a likelihood of not less than 0.3 across all therapeutic alternatives.

$\varphi_8 = Q_{>0.5}(f\oplus_{0.4}e)$ encapsulates the fulfillment condition for the possibility that the patient's health status will consistently comply with the requirement of being "fine" with a 40% weightage and "excellent" with a 60% weightage, with a likelihood exceeding 0.5 across all therapeutic alternatives.

$\varphi_9 = Q_{>0.6}(0.5_{ne}(f)\sqcup e)$ delineates the fulfillment condition for the possibility that the patient's health status will remain "fine" with a necessity not exceeding 0.5 until it transitions to "excellent," with a likelihood exceeding 0.6 across all therapeutic alternatives.

It is abundantly clear that these temporal properties with quality constraints exceed the expressive capabilities of PoCTL.

3. The Relations and Attributes of QFCTL Formulas.

Initially, we delineate the notion of relationships between QFCTL formulas.

Definition 7 [Relationships Between QFCTL Formulas] Let φ, φ_1, φ_2 denote QFCTL state formulas and ψ, ψ_1, ψ_2 denote QFCTL path formulas. *Consider $M = (S, I, \delta, AP, L)$ to be a FKS, where $Path(M) = \{\pi|\pi \in S^{\omega}, I(\pi_0) > 0\}$ is the set of infinite paths in M. Let "$\sim$" represent a relational operator, which may be selected from the set $\{>, <, \geq, \leq, \neq, =\}$. The relations among QFCTL formulas are delineated as follows:*

(1) If for every state $s \in S$, the evaluation $[\![s, \varphi_1]\!] \sim [\![s, \varphi_2]\!]$, then φ_1 is deemed to be in relation "$\sim$" with φ_2, indicated as $\varphi_1 \sim \varphi_2$.

(2) If for every path $\pi \in Path(M)$, the evaluation $[\![\pi, \psi_1]\!] \sim [\![\pi, \psi_2]\!]$, then ψ_1 is deemed to be in relation "$\sim$" with ψ_2, indicated as $\psi_1 \sim \psi_2$.

Subsequently, our research expedition advances towards the meticulous examination of the distinctive logical attributes intrinsic to Quantified Fuzzy Computation Tree Logic (QFCTL). As a pivotal pillar within its own theoretical architecture, this in-depth exploration is paramount for enhancing comprehension of the fundamental logical mechanisms that underpin this specialized field.

Theorem 1 [Equivalence Calculus of QFCTL Formulas] *Let* φ, φ_1, φ_2 *be state formulas in QFCTL, and* $M = (S, I, \delta, AP, L)$ *be an FKS. Then the following conclusions hold:*

(1) $\varphi = \neg\neg\varphi$;

(2) $\lambda_{cp}(\varphi) = \neg\lambda_{ne}(\neg\varphi)$, $\lambda_{ne}(\varphi) = \neg\lambda_{cp}(\neg\varphi)$;

(3) $\lambda_{ne}(\varphi) = \varphi \oplus_{\lambda} Ture$, $\lambda_{cf}(\varphi) = \lambda_{cp}(\varphi) + (1-\lambda)/2$;

(4) $\varphi_1 \oplus_{\lambda} \varphi_2 = \varphi_2 \oplus_{1-\lambda} \varphi_1$;

(5) $\varphi \oplus_{\lambda} (\varphi_1 \wedge \varphi_2) = (\varphi \oplus_{\lambda} \varphi_1) \wedge (\varphi \oplus_{\lambda} \varphi_2)$, $\varphi \oplus_{\lambda} (\varphi_1 \vee \varphi_2) = (\varphi \oplus_{\lambda} \varphi_1) \vee (\varphi \oplus_{\lambda} \varphi_2)$;

(6) $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$;

(7) $\varphi_1 \longrightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$;

(8) $\Diamond\varphi = Ture \sqcup \varphi$;

(9) $\exists\Diamond\varphi = \exists(Ture \sqcup \varphi)$, $\forall\Diamond\varphi = \forall(Ture \sqcup \varphi)$;

(10) $\exists\bigcirc\varphi = \neg\forall\bigcirc\neg\varphi$, $\forall\bigcirc\varphi = \neg\exists\bigcirc\neg\varphi$;

(11) $\exists\Box\varphi = \neg\forall\Diamond\neg\varphi$, $\forall\Box\varphi = \neg\exists\Diamond\neg\varphi$;

(12) $\bigcirc\varphi = \neg(\bigcirc\neg\varphi)$, $\Box\varphi = \neg(\Diamond\neg\varphi)$, $\Diamond\varphi = \neg(\Box\neg\varphi)$;

(13) $\Diamond\varphi = \varphi \vee \bigcirc\Diamond\varphi$, $\Box\varphi = \varphi \wedge \bigcirc\Box\varphi$;

(14) $\varphi_1 \sqcup \varphi_2 = \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \sqcup \varphi_2)$;

(15) $\forall\Box(\varphi_1 \wedge \varphi_2) = \forall\Box\varphi_1 \wedge \forall\Box\varphi_2$;

(16) $\exists\Box(\varphi_1 \vee \varphi_2) = \exists\Box\varphi_1 \vee \exists\Box\varphi_2$;

Proof: The above conclusions can be readily proven through straightforward calculations based on Definition 6. Here, we provide proof examples for (2), (5), (14), (15), and (16) as follows.

Proof of Conclusion (2). For all $s \in S$,

$\llbracket s, \neg\lambda_{ne}(\neg\varphi) \rrbracket = 1 - \llbracket s, \lambda_{ne}(\neg\varphi) \rrbracket$

$= 1 - ((1-\lambda) + \lambda\llbracket s, \neg\varphi \rrbracket)\ \lambda_{cp}(\varphi)$

$= 1 - \big((1-\lambda) + \lambda(1 - \llbracket s, \varphi \rrbracket)\big)$

$= \lambda\llbracket s, \varphi \rrbracket = \llbracket s, \lambda_{cp}(\varphi) \rrbracket$.

Therefore, $\lambda_{cp}(\varphi) = \neg\lambda_{ne}(\neg\varphi)$. By analogous reasoning, it can be shown that,

$$\lambda_{ne}(\varphi) = \neg\lambda_{cp}(\neg\varphi).$$

Proof of Conclusion (5). For all $s \in S$,

$\llbracket s, \varphi \oplus_{\lambda} (\varphi_1 \wedge \varphi_2) \rrbracket = \lambda\llbracket s, \varphi \rrbracket + (1-\lambda)(\llbracket s, \varphi_1 \rrbracket \wedge \llbracket s, \varphi_2 \rrbracket)$

$= (\lambda\llbracket s, \varphi \rrbracket + (1-\lambda)\llbracket s, \varphi_1 \rrbracket) \wedge (\lambda\llbracket s, \varphi \rrbracket + (1-\lambda)\llbracket s, \varphi_2 \rrbracket)$

$= \llbracket s, \varphi \oplus_{\lambda} \varphi_1 \rrbracket \wedge \llbracket s, \varphi \oplus_{\lambda} \varphi_1 \rrbracket$.

Therefore, $\varphi \oplus_{\lambda} (\varphi_1 \wedge \varphi_2) = (\varphi \oplus_{\lambda} \varphi_1) \wedge (\varphi \oplus_{\lambda} \varphi_2)$. By analogous reasoning, it can be shown that, $\varphi \oplus_{\lambda} (\varphi_1 \vee \varphi_2) = (\varphi \oplus_{\lambda} \varphi_1) \vee (\varphi \oplus_{\lambda} \varphi_2)$.

Proof of Conclusion (14). For all $\pi \in Path(M)$, $i \in \mathbb{N}$,

$\llbracket \pi^i, \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \sqcup \varphi_2)) \rrbracket$

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee \llbracket \pi^i, \varphi_1 \wedge \bigcirc(\varphi_1 \sqcup \varphi_2) \rrbracket$

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee (\llbracket \pi^i, \varphi_1 \rrbracket \wedge \llbracket \pi^i, \bigcirc(\varphi_1 \sqcup \varphi_2) \rrbracket)$

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee (\llbracket \pi^i, \varphi_1 \rrbracket \wedge \llbracket \pi^{i+1}, \varphi_1 \sqcup \varphi_2 \rrbracket)$

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee (\llbracket \pi^i, \varphi_1 \rrbracket \wedge \bigvee_{j \geq i+1} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge \bigwedge_{i+1 \leq k < j} \llbracket \pi^k, \varphi_1 \rrbracket))$

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee \bigvee_{j \geq i+1} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge (\llbracket \pi^i, \varphi_1 \rrbracket \wedge \bigwedge_{i+1 \leq k < j} \llbracket \pi^k, \varphi_1 \rrbracket))$

doi:10.20944/preprints202410.1921.v1

$= \llbracket \pi^i, \varphi_2 \rrbracket \vee \bigvee_{j \ge i+1} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge \bigwedge_{i \le k < j} \llbracket \pi^k, \varphi_1 \rrbracket)$
$= (\llbracket \pi^i, \varphi_2 \rrbracket \wedge Ture) \vee \bigvee_{j \ge i+1} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge \bigwedge_{i \le k < j} \llbracket \pi^k, \varphi_1 \rrbracket)$
$= (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{i \le k < i} \llbracket \pi^k, \varphi_1 \rrbracket) \vee \bigvee_{j \ge i+1} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge \bigwedge_{i \le k < j} \llbracket \pi^k, \varphi_1 \rrbracket)$
$= \bigvee_{j \ge i} (\llbracket \pi^j, \varphi_2 \rrbracket \wedge \bigwedge_{i \le k < j} \llbracket \pi^k, \varphi_1 \rrbracket)$
$= \llbracket \pi^i, \varphi_1 \sqcup \varphi_2 \rrbracket$.

Conclusion (14) Established.

Proof of Conclusion (15). For all $s \in S$,
$\llbracket s, \forall \Box (\varphi_1 \wedge \varphi_2) \rrbracket = \bigwedge_{\pi \in Path(s)} \llbracket \pi, \Box (\varphi_1 \wedge \varphi_2) \rrbracket$
$= \bigwedge_{\pi \in Path(s)} \bigwedge_{i \ge 0} \llbracket \pi^i, \varphi_1 \wedge \varphi_2 \rrbracket$
$= \bigwedge_{\pi \in Path(s)} (\bigwedge_{i \ge 0} \llbracket \pi^i, \varphi_1 \rrbracket \wedge \bigwedge_{i \ge 0} \llbracket \pi^i, \varphi_2 \rrbracket)$
$= (\bigwedge_{\pi \in Path(s)} \bigwedge_{i \ge 0} \llbracket \pi^i, \varphi_1 \rrbracket) \wedge (\bigwedge_{\pi \in Path(s)} \bigwedge_{i \ge 0} \llbracket \pi^i, \varphi_2 \rrbracket)$
$= \bigwedge_{\pi \in Path(s)} \llbracket \pi, \Box \varphi_1 \rrbracket \wedge \bigwedge_{\pi \in Path(s)} \llbracket \pi, \llbracket \pi, \Box \varphi_2 \rrbracket \rrbracket$
$= \llbracket s, \forall \Box \varphi_1 \rrbracket \wedge \llbracket s, \forall \Box \varphi_2 \rrbracket$
$= \llbracket s, \forall \Box \varphi_1 \wedge \forall \Box \varphi_2 \rrbracket$.

Conclusion (15) Established.

The proof for Conclusion (16) is identical to that of Conclusion (15) and will not be repeated here. This completes the proof of Theorem 1. □

Proposition 1. *The set* $\{\neg, \vee, \oplus_\lambda, Q_P(\cdot), \exists \bigcirc, \exists \sqcup, \forall \sqcup\}$ *constitutes a functionally complete set of operators for QFCTL.*

With the functionally complete set of operators for QFCTL, we only need to consider the operators within this set during model checking.

Proposition 2 [Inequality Relations Among QFCTL Formulas] *Let* $M = (S, I, \delta, AP, L)$ *be an FKS,* φ *signifies a QFCTL state formula, and* ψ *denotes a QFCTL path formula. Then,*

(1) $\lambda_{cp}(\varphi) \le \lambda_{cf}(\varphi) \le \lambda_{ne}(\varphi)$, *with equality holding if and only if* $\lambda = 1$;
(2) $\Box \varphi \le \bigcirc \varphi \le \Diamond \varphi$;
(3) $\forall \psi \le \exists \psi$.

The aforementioned conclusions can be readily proven by simple calculations according to Definition 6.

The characteristic predicate operator $Q_P(\cdot)$ qualitatively characterizes the relationship between the satisfaction value of a QFCTL formula and a quality predicate $P \subseteq [0,1] \cap \mathbb{Q}$. Below are the properties and decision theorems for the characteristic predicate operator.

Theorem 2 [Fundamental Properties and Decisions of the Characteristic Predicate Operator] *Let* $= (S, I, \delta, AP, L)$ *be an FKS,* ψ *denotes a QFCTL path formula. Let* $P = [\alpha, \beta] \subseteq [0,1]$ *be a quality predicate, where* $\bar{P}$ *denotes the complement of* P *in [0,1], and* $v[s, \psi] = \{\llbracket \pi, \psi \rrbracket | \pi \in Path(s)\}$ *represents the set of satisfaction values of* ψ *over all paths starting from state* $s \in S$. *Then,*

(1) $Q_P(\psi) + Q_{\bar{P}}(\psi) = 1, Q_P(\neg\psi) = Q_{\bar{P}}(\psi)$;
(2) $Q_P(\psi) = 0$ *if and only if* $\forall \psi > \beta$ *or* $\exists \psi < \alpha$;
(3) $Q_P(\psi) = 1$ *if and only if* $\alpha \le \forall \psi \le \exists \psi \le \beta$;
(4) $Q_P(\psi) = 0.5$ *if and only if* $\forall \psi < \alpha \le \exists \psi \le \beta$ *or* $\alpha \le \forall \psi \le \beta < \exists \psi$.

Proof: (1) For all $s \in S$, if $\llbracket s, Q_P(\psi) \rrbracket = 1$, then for all $\pi \in Path(s)$, $\llbracket \pi, \psi \rrbracket \in P$. Consequently, for all $\pi \in Path(s)$, $\llbracket \pi, \psi \rrbracket \notin \bar{P}$, where $\bar{P}$ denotes the complement of P. This implies $\llbracket s, Q_{\bar{P}}(\psi) \rrbracket = 0$, so we have $Q_P(\psi) + Q_{\bar{P}}(\psi) = 1$. Conversely, if $\llbracket s, Q_P(\psi) \rrbracket = 0$, then it follows that $\llbracket s, Q_{\bar{P}}(\psi) \rrbracket = 1$, again yielding $Q_P(\psi) + Q_{\bar{P}}(\psi) = 1$.

For all $s \in S$, if $\llbracket s, Q_P(\psi) \rrbracket = 0.5$, then there exist $\pi, \pi' \in Path(s), \llbracket \pi, \psi \rrbracket \in P$ and $\llbracket \pi', \psi \rrbracket \notin P$ simultaneously. Consequently, $\llbracket s, Q_{\bar{P}}(\psi) \rrbracket = 0.5$. Therefore, $Q_P(\psi) + Q_{\bar{P}}(\psi) = 1$.

This demonstrates that the sum of the predicates $Q_P(\psi)$ and $Q_{\bar{P}}(\psi)$ is always unity, reflecting the complementary nature of the sets P and $\bar{P}$ within the logical framework defined.

For all $s \in S$, if $[\![s, Q_P(\neg\psi)]\!] = 1$, then for all $\pi \in Path(s)$, $[\![\pi, \neg\psi]\!] \in P$. Consequently, for all $\pi \in Path(s)$, $1 - [\![\pi, \psi]\!] \in P$, which implies $[\![\pi, \psi]\!] \in \bar{P}$. Thus, $\left[\!\left[s, Q_{\bar{P}}(\psi)\right]\!\right] = 1 = \left[\!\left[s, Q_P(\neg\psi)\right]\!\right]$. Conversely, if $[\![s, Q_P(\neg\psi)]\!] = 0$, then $[\![s, Q_{\bar{P}}(\psi)]\!] = 0$, and also $[\![s, Q_{\bar{P}}(\psi)]\!] = 0 = [\![s, Q_P(\neg\psi)]\!]$.

For all $s \in S$, if $[\![s, Q_P(\neg\psi)]\!] = 0.5$, then there exist $\pi, \pi' \in Path(s), [\![\pi, \neg\psi]\!] \in P$ and $[\![\pi', \neg\psi]\!] \notin P$. This implies that $[\![\pi, \psi]\!] \in \bar{P}$ and $[\![\pi', \psi]\!] \notin \bar{P}$. Hence $[\![s, Q_{\bar{P}}(\psi)]\!] = 0.5$. Therefore, $[\![s, Q_{\bar{P}}(\psi)]\!] = 0.5 = [\![s, Q_P(\neg\psi)]\!]$.

(2) First, prove the sufficiency. When $\forall\psi > \beta$, for all $s \in S$, since $\beta < [\![s, \forall\psi]\!] = \bigwedge_{\pi \in Path(s)} [\![\pi, \psi]\!]$, it follows that for all $\pi \in Path(s), [\![\pi, \psi]\!] > \beta$. Given that β is the upper bound of $P = [\alpha, \beta]$, it implies that for all $\pi \in Path(s)$, $[\![\pi, \psi]\!] \notin P$. According to Definition 6, we conclude that $[\![s, Q_P(\psi)]\!] = 0$. This intuitive result is illustrated in Figure 2.

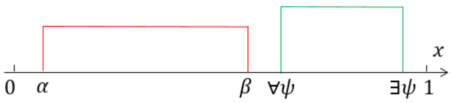

Figure 2. $Q_P(\psi) = 0$ when $\forall\psi > \beta$.

When $\exists\psi < \alpha$, for all $s \in S$, since $\alpha > [\![s, \exists\psi]\!] = \bigvee_{\pi \in Path(s)} [\![\pi, \psi]\!]$, it follows that for all $\pi \in Path(s), [\![\pi, \psi]\!] < \alpha$. Given that α is the lower bound of $P = [\alpha, \beta]$,, it implies that for all $\pi \in Path(s), [\![\pi, \psi]\!] \notin P$. According to Definition 6, we conclude that $[\![s, Q_P(\psi)]\!] = 0$. This intuitive result is illustrated in Figure 3.

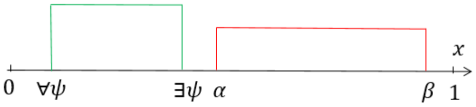

Figure 3. $Q_P(\psi) = 0$ when $\exists\psi < \alpha$.

In fact, (2) corresponds to $v[s, \psi] \cap P = \Phi$, which means that $v[s, \psi]$ and P are disjoint. Conversely, when proving the necessity, $Q_P(\psi) = 0$ corresponds to two scenarios depicted in Figure 2 and Figure 3, leading straightforwardly to the conclusion that either $\forall\psi > \beta$ *or* $\exists\psi < \alpha$. Thus, conclusion (2) is proven.

(3) First, prove the sufficiency. When $\alpha \le \forall\psi \le \exists\psi \le \beta$, for all $s \in S$, on one hand, we have $\alpha \le [\![s, \forall\psi]\!] = \bigwedge_{\pi \in Path(s)} [\![\pi, \psi]\!]$, implying that for all $\pi \in Path(s), [\![\pi, \psi]\!] \ge \alpha$. On the other hand, $\beta \ge [\![s, \exists\psi]\!] = \bigvee_{\pi \in Path(s)} [\![\pi, \psi]\!]$, implying that for all $\pi \in Path(s), [\![\pi, \psi]\!] \le \beta$. Consequently, for all $\pi \in Path(s), [\![\pi, \psi]\!] \in P$. According to Definition 6, we conclude that $[\![s, Q_P(\psi)]\!] = 1$. This intuitive result is illustrated in Figure 4. In fact, (3) corresponds to $v[s, \psi] \subseteq P$, i.e., $v[s, \psi]$ is contained within P.

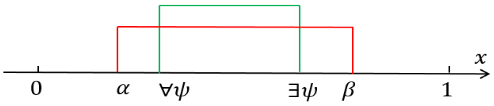

Figure 4. $Q_P(\psi) = 1$ when $\alpha \le \forall\psi \le \exists\psi \le \beta$.

With reference to Figure 4, it is straightforward to demonstrate the necessity of conclusion (3). Thus, conclusion (3) is proven.

(4) First, prove the sufficiency. When $\forall\psi < \alpha \le \exists\psi \le \beta$, for all $s \in S$, on one hand, $\alpha > [\![s, \forall\psi]\!] = \bigwedge_{\pi \in Path(s)} [\![\pi, \psi]\!]$, which implies that there exists $\pi \in Path(s)$, so that $[\![\pi, \psi]\!] \notin P$. On the other hand, from $\alpha \le [\![s, \exists\psi]\!] \le \beta$, we know that $\alpha \le \bigvee_{\pi \in Path(s)} [\![\pi, \psi]\!] \le \beta$, which indicates that there exists

$\pi', \in Path(s)$, so that $\alpha \leq [\![\pi', \psi]\!] \leq \beta$. . Therefore, there exists $\pi', \in Path(s)$, so that $[\![\pi', \psi]\!] \in P$. According to Definition 6, we conclude that $[\![s, Q_P(\psi)]\!] = 0.5$. This intuitive result is illustrated in Figure 5.

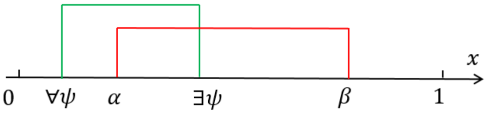

Figure 5. $Q_P(\psi) = 0.5$ when $\forall\psi < \alpha \leq \exists\psi \leq \beta$.

When $\alpha \leq \forall\psi \leq \beta < \exists\psi$, similarly, it can be proven that $Q_P(\psi) = 0.5$. This is intuitively illustrated in Figure 6.

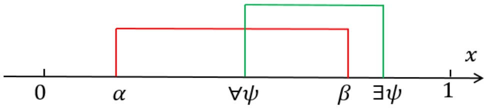

Figure 6. $Q_P(\psi) = 0.5$ when $\alpha \leq \forall\psi \leq \beta < \exists\psi$.

In fact, (4) corresponds to the case where $v[s, \psi]$ partially intersects with P. With reference to Figures 5 and 6, the proof of the necessity of conclusion (4) is also evident.

Proof of Theorem 4 is complete. □

Corollary 1. *Let $M = (S, I, \delta, AP, L)$ be an FKS, and ψ be a QFCTL path formula. When the quality predicate P is an open or half-open interval, the conclusion (1) of Theorem 2 holds. The conditions for conclusions (2), (3), and (4) can be slightly modified to yield the following conclusions:*

When $P = (\alpha, \beta) \subset [0,1]$, we have:
(1) $Q_P(\psi) = 0$ *if and only if* $\forall\psi \geq \beta$ *or* $\exists\psi \leq \alpha$;
(2) $Q_P(\psi) = 1$ *if and only if* $\alpha < \forall\psi \leq \exists\psi < \beta$;
(3) $Q_P(\psi) = 0.5$ *if and only if* $\forall\psi \leq \alpha < \exists\psi < \beta$ *or* $\alpha < \forall\psi < \beta \leq \exists\psi$.
When $P = (\alpha, \beta] \subset [0,1]$, we have:
(4) $Q_P(\psi) = 0$ *if and only if* $\forall\psi > \beta$ *or* $\exists\psi \leq \alpha$;
(5) $Q_P(\psi) = 1$ *if and only if* $\alpha < \forall\psi \leq \exists\psi \leq \beta$;
(6) $Q_P(\psi) = 0.5$ *if and only if* $\forall\psi \leq \alpha < \exists\psi \leq \beta$ *or* $\alpha < \forall\psi \leq \beta < \exists\psi$.
When $P = [\alpha, \beta) \subset [0,1]$, we have:
(7) $Q_P(\psi) = 0$ *if and only if* $\forall\psi \geq \beta$ *or* $\exists\psi < \alpha$;
(8) $Q_P(\psi) = 1$ *if and only if* $\alpha \leq \forall\psi \leq \exists\psi < \beta$;
(9) $Q_P(\psi) = 0.5$ *if and only if* $\forall\psi < \alpha \leq \exists\psi < \beta$ *or* $\alpha \leq \forall\psi < \beta \leq \exists\psi$.

From Theorem 2 and Corollary 1, we know that the values of $[\![s, \exists\psi]\!]$ and $[\![s, \forall\psi]\!]$ can be computed first, and then the value of $[\![s, Q_P(\psi)]\!]$ can be derived.

4. Model Checking for QFCTL

The semantics of QFCTL are defined on FKS (Fuzzy Kripke Structures), incorporating quality constraint operators and characteristic predicate operators. Compared to PoCTL model checking, QFCTL poses additional challenges in three key areas. Firstly, when computing the semantics of formulas, it integrates information from both path reachability degrees and satisfaction values of system properties. This integration is not a binary choice but a synthesis based on different preference requirements, undoubtedly increasing computational complexity. Secondly, the integration of these two types of information necessitates constant synchronization, meaning that the satisfaction value of a property on a specific path segment should be combined with the reachability degree of that segment, rather than aggregating the satisfaction value of the property over the entire path and then combining it with the overall reachability degree of the path. This synchronization requirement further elevates computational difficulty. Lastly, quality constraints are applied to subformulas of properties to reflect the importance and preferences of different properties, which also adds to the computational complexity. Therefore, explicit model checking for QFCTL cannot be conducted using fuzzy matrix composition operations[20,21], as these operations are incompatible with quality constraint computations. Fortunately, FKS are finite, with limited numbers of atomic propositions, propositional constants, and quality constraint operators. By leveraging ideas from fixed-point theory[18,19], we reduce the semantic computation of QFCTL on infinite paths to iterative calculations within a finite number of steps.

4.1. Description of the Model Checking Problem for QFCTL

Analogous to the model checking problem for PoCTL, the model checking problem for QFCTL can be described as follows. Let φ be a QFCTL state formula, and $M = (S, I, \delta, AP, L)$ be an FKS (Fuzzy Kripke Structure). Let $P \subseteq [0,1] \cap \mathbb{Q}$ denote the quality predicate, and $APR \subseteq [0,1] \cap \mathbb{Q}$ be a finite set of fuzzy constants. The model checking problem for QFCTL involves computing the value of $[\![s, \varphi]\!]$ for all $s \in S$ such that $I(s) > 0$. For QFCTL, the model checking problem is extended to compute the truth values of formulas that incorporate quality constraint functions and characteristic predicates, thus necessitating a different model checking approach.

From Proposition 1, it is known that$\{\neg, \vee, \oplus_\lambda, Q_P(\cdot), \exists\bigcirc, \exists \sqcup, \forall \sqcup\}$ constitutes a functionally complete set of operators for QFCTL formulas. According to Definition 6, the operators $\neg, \vee$ $, \oplus_\lambda$ and $\exists\bigcirc$ can be directly evaluated on the current state or the current state along with its immediate successor state to obtain solutions. Therefore, it suffices to provide algorithms for solving QFCTL formulas with the operators $Q_P(\cdot), \exists \sqcup,$ and $\forall \sqcup$. Theorems 2 and Corollary 1 can serve as decision theorems for model checking with the operator $Q_P(\cdot)$. For any path formula ψ and quality predicate P, by solving for $[\![s, \exists\psi]\!]$ and $[\![s, \forall\psi]\!]$ for all $s \in S$, we can obtain the value of $[\![s, Q_P(\psi)]\!]$. Hence, the core problem of QFCTL model checking lies in computing the truth values of QFCTL formulas with the operators $\exists \sqcup$ and $\forall \sqcup$. We reduce the computation of these two operators to finite paths, and the QFCTL model checking problem can be solved through a finite number of iterations.

For all $n \in \mathbb{N}$, the semantics of the "bounded until" operator are defined as follows:
$[\![\pi, \varphi_1 \sqcup^n \varphi_2]\!] = \bigvee_{0 \le i < n} ([\![\pi^i, \varphi_2]\!] \wedge \bigwedge_{0 \le j < i} [\![\pi^j, \varphi_1]\!])$
$= \bigvee_{0 \le i < n} (\delta_i^*(\pi) \circ [\![\pi_i, \varphi_2]\!]) \wedge \bigwedge_{0 \le j < i} (\delta_j^*(\pi) \circ [\![\pi_j, \varphi_1]\!])).$

The following theorem provides the basis for iteratively solving the model checking problem for QFCTL formulas with the operators $\exists \sqcup$ and $\forall \sqcup$ within a finite number of steps.

Theorem 3. *Let φ_1 and φ_2 be QFCTL state formulas, and $M = (S, I, \delta, AP, L)$ be an FKS. Then the following conclusions hold:*
(1) $\exists(\varphi_1 \sqcup \varphi_2) = \exists(\varphi_1 \sqcup^{|S|} \varphi_2)$;
(2) $\forall(\varphi_1 \sqcup \varphi_2) = \forall(\varphi_1 \sqcup^{|S|} \varphi_2)$;
(3) $\exists(\varphi_1 \sqcup \varphi_2) = \varphi_2 \vee (\varphi_1 \wedge \exists\bigcirc\exists(\varphi_1 \sqcup \varphi_2)$;
(4) $\forall(\varphi_1 \sqcup \varphi_2) = \varphi_2 \vee (\varphi_1 \wedge \forall\bigcirc\forall(\varphi_1 \sqcup \varphi_2)$.

Proof: (1) It is naturally true that $\exists(\varphi_1 \sqcup^{|S|} \varphi_2) \le \exists(\varphi_1 \sqcup \varphi_2)$. What needs to be proven is that $\exists(\varphi_1 \sqcup \varphi_2) \le \exists(\varphi_1 \sqcup^{|S|} \varphi_2)$.

We prove that for all $s \in S$ and for all $\pi \in Path(s)$, it holds that,

$$[\![\pi, \varphi_1 \sqcup \varphi_2]\!] \le [\![s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2)]\!]. \tag{1}$$

It suffices to prove that for all $k \in \mathbb{N}$,

$$[\![\pi, \varphi_1 \sqcup^k \varphi_2]\!] \le [\![s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2)]\!]. \tag{2}$$

When $k < |S|$, the Inequality (2) obviously holds. When $k = |S|$, $[\![\pi, \varphi_1 \sqcup^{|S|} \varphi_2]\!] \le [\![s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2)]\!]$,the Inequality (2) also holds. Now, assume that the Inequality (2) holds when $k = m \ge |S|$, i.e.,

$$[\![\pi, \varphi_1 \sqcup^m \varphi_2]\!] \le [\![s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2)]\!]. \tag{3}$$

We prove that the inequality (2) holds when $k = m + 1$. The discussion is divided into cases as follows:

If there exists $0 \le t < m$ where $\pi_m = \pi_t$, then,

$$\llbracket \pi, \varphi_1 \sqcup^{m+1} \varphi_2 \rrbracket = \bigvee_{0 \le i < m+1} \Big(\bigwedge_{0 \le j < i} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_i^*(\pi) \circ \llbracket \pi_i, \varphi_2 \rrbracket \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \Big(\bigwedge_{0 \le j < m} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_m^*(\pi) \circ \llbracket \pi_m, \varphi_2 \rrbracket \Big)$$
$$\le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \Big(\bigwedge_{0 \le j < m} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_t^*(\pi) \circ \llbracket \pi_t, \varphi_2 \rrbracket \Big)$$
$$\le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \Big(\bigwedge_{0 \le j < t} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_t^*(\pi) \circ \llbracket \pi_t, \varphi_2 \rrbracket \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket.$$

Therefore, applying the induction hypothesis (Inequality (3)), we have

$$\llbracket \pi, \varphi_1 \sqcup^{m+1} \varphi_2 \rrbracket \le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \le \llbracket s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2) \rrbracket.$$

If for all $t < m$, where $\pi_m \ne \pi_t$, then since $m + 1 > |S|$, there must exist $0 < l < h < m$ on the path $\pi = \pi_0, \pi_1, \cdots, \pi_m, \cdots$, such that $\pi_l = \pi_h$, creating a loop. By removing the path segment $\pi_{l+1}, \cdots, \pi_h$, we construct a new path $\rho = \pi_0, \pi_1, \cdots, \pi_l, \cdots \pi_{h+1}, \cdots, \pi_m, \cdots$. Relabeling in order, we obtain $\rho = \rho_0, \rho_1, \cdots, \rho_{m-(h-l)}, \cdots$, where for all $i \le l, \rho_i = \pi_i$, and for all $i > h$ $\rho_{i-(h-l)} = \pi_i$.

$$\llbracket \pi, \varphi_1 \sqcup^{m+1} \varphi_2 \rrbracket = \bigvee_{0 \le i < m+1} \Big(\bigwedge_{0 \le j < i} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_i^*(\pi) \circ \llbracket \pi_i, \varphi_2 \rrbracket \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigwedge_{0 \le j < m} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_m^*(\pi) \circ \llbracket \pi_m, \varphi_2 \rrbracket \Big)$$
$$\le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigwedge_{0 \le j < m-(h-l)} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_m^*(\pi) \circ \llbracket \pi_m, \varphi_2 \rrbracket \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigwedge_{0 \le j < m-(h-l)} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_m^*(\pi) \circ \llbracket \rho_{m-(h-l)}, \varphi_2 \rrbracket \Big)$$
$$\le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigwedge_{0 \le j < m-(h-l)} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_{m-(h-l)}^*(\pi) \circ \llbracket \rho_{m-(h-l)}, \varphi_2 \rrbracket \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigwedge_{0 \le j < m-(h-l)} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_{m-(h-l)}^*(\rho) \circ \llbracket \rho_{m-(h-l)}, \varphi_2 \rrbracket \Big)$$
$$\le \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \bigvee_{0 \le i < m-(h-l)+1} \Big(\bigwedge_{0 \le j < i} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_i^*(\rho) \circ \llbracket \rho_i, \varphi_2 \rrbracket \Big) \Big)$$
$$= \llbracket \pi, \varphi_1 \sqcup^{m} \varphi_2 \rrbracket \vee \llbracket \rho, \varphi_1 \sqcup^{m-(h-l)+1} \varphi_2 \rrbracket.$$

Since $0 < l < h < m$, it follows that $m - (h - l) + 1 \le m$. According to the induction hypothesis (Inequality (3)), we have,

$$\llbracket \rho, \varphi_1 \sqcup^{m-(h-l)+1} \varphi_2 \rrbracket \le \llbracket s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2) \rrbracket. \tag{4}$$

Combining Inequality (3) and (4), we obtain,

$$\llbracket \pi, \varphi_1 \sqcup^{m+1} \varphi_2 \rrbracket \le \llbracket s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2) \rrbracket.$$

Thus, Inequality (2) holds, for all $k \in \mathbb{N}$, $\llbracket \pi, \varphi_1 \sqcup^{k} \varphi_2 \rrbracket \le \llbracket s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2) \rrbracket$. Therefore,

$$\llbracket \pi, \varphi_1 \sqcup \varphi_2 \rrbracket \le \llbracket s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2) \rrbracket.$$

This completes the proof of Conclusion (1).

The proof of Conclusion (2) is analogous to the proof of (1) and will not be repeated here.

Next, we proceed to prove Conclusion (3). For all $s \in S$,

$$\llbracket s, \varphi_2 \vee (\varphi_1 \wedge \exists \bigcirc \exists(\varphi_1 \sqcup \varphi_2)) \rrbracket$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\llbracket s, \varphi_1 \rrbracket \wedge \llbracket s, \exists \bigcirc \exists(\varphi_1 \sqcup \varphi_2) \rrbracket$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\llbracket s, \varphi_1 \rrbracket \wedge \bigvee_{\pi \in Path(s)} \llbracket \pi, \bigcirc \exists(\varphi_1 \sqcup \varphi_2) \rrbracket)$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\llbracket s, \varphi_1 \rrbracket \wedge \bigvee_{\pi \in Path(s)} (\llbracket \pi_1, \exists(\varphi_1 \sqcup \varphi_2) \rrbracket))$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\llbracket s, \varphi_1 \rrbracket \wedge \bigvee_{\pi \in Path(s)} (\bigvee_{\rho \in Path(\pi_1)} \llbracket \rho, \varphi_1 \sqcup \varphi_2 \rrbracket))$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\llbracket s, \varphi_1 \rrbracket \wedge \bigvee_{\pi \in Path(s)} (\bigvee_{\rho \in Path(\pi_1)} (\bigvee_{0 \le i < |S|} (\llbracket \rho^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \rho^j, \varphi_1 \rrbracket))))$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\bigvee_{\pi \in Path(s)} (\bigvee_{\rho \in Path(\pi_1)} (\bigvee_{0 \le i < |S|} (\llbracket \rho^i, \varphi_2 \rrbracket \wedge \llbracket s, \varphi_1 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \rho^j, \varphi_1 \rrbracket))))$$
$$= \llbracket s, \varphi_2 \rrbracket \vee (\bigvee_{\pi \in Path(s)} (\bigvee_{1 \le i < |S|+1} (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \pi^j, \varphi_1 \rrbracket))$$
$$= \bigvee_{\pi \in Path(s)} (\llbracket \pi^0, \varphi_2 \rrbracket \vee \bigvee_{1 \le i < |S|+1} (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \pi^j, \varphi_1 \rrbracket))$$
$$= \bigvee_{\pi \in Path(s)} ((\llbracket \pi^0, \varphi_2 \rrbracket \wedge 1) \vee \bigvee_{1 \le i < |S|+1} (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \pi^j, \varphi_1 \rrbracket))$$
$$= \bigvee_{\pi \in Path(s)} ((\llbracket \pi^0, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < 0} \llbracket \pi^j, \varphi_1 \rrbracket) \vee \bigvee_{1 \le i < |S|+1} (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \pi^j, \varphi_1 \rrbracket))$$
$$= \bigvee_{\pi \in Path(s)} (\bigvee_{0 \le i < |S|+1} (\llbracket \pi^i, \varphi_2 \rrbracket \wedge \bigwedge_{0 \le j < i} \llbracket \pi^j, \varphi_1 \rrbracket))$$
$$= \bigvee_{\pi \in Path(s)} \llbracket \pi, \varphi_1 \sqcup^{|S|+1} \varphi_2 \rrbracket$$

$$= \bigvee_{\pi \in Path(s)} [\![\pi, \varphi_1 \sqcup^{|S|} \varphi_2]\!] \quad \text{(Based on Inequality (2))}$$
$$= [\![s, \exists(\varphi_1 \sqcup^{|S|} \varphi_2)]\!]$$
$$= [\![s, \exists(\varphi_1 \sqcup \varphi_2)]\!]. \quad \text{(Based on Conclusion (1))}$$

The proof of Conclusion (4) is similar to the proof of Conclusion (3) and will not be repeated here. This completes the proof of Theorem 3. □

4.2. Solution Algorithms and Complexity Analysis for the Model Checking Problem of QFCTL

Below, we present the solution algorithm for the model checking problem of the QFCTL formula $\exists(\varphi_1 \sqcup \varphi_2)$.

Algorithm 1 The Computation Algorithm of $\exists(\varphi_1 \sqcup \varphi_2)$.

Input: A QFCTL formula $\exists(\varphi_1 \sqcup \varphi_2)$, a FKS $M = (S, I, \delta, AP, L)$, with the information fusion operator $\circ \in \{\wedge, \cdot, \oplus_\lambda\}$.

Solution process:

```
1 Initialization: Fuzzy set $v, v': S \longrightarrow [0,1]$, $\forall s \in S$, $v(s) = [\![s, \varphi_2]\!]$, $v'(s) = [\![s, \varphi_1]\!]$.
2 LOOP $\forall s \in S, I(s) > 0$
3     $\delta^*(s) = I(s)$;
4     $f(s) = \delta^*(s)^\circ v(s)$;// Initialize the objective function value for the source state $s$.
5     $\mu = s$;// Label the current expanded state.
6     LOOP from $i = 1$ to $i = |S|$
7         $f(\mu) = f(\mu) \vee \left(\delta^*(\mu)^\circ v'(\mu) \wedge \bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ v(s')\right)$; // Update the object tive function value for the current state $\mu$.
8         $\delta^*(\mu) = \delta^*(\mu) \wedge \delta(\mu, s')$;// Update the reachability of the path.
9         $f(s) = f(\mu)$;// Update the objective function value for the source state $s$.
10        $\mu = s'$;// Update the current state.
11    END
12 END
```

Output: Fuzzy set f.

When computing $[\![s, \forall(\varphi_1 \sqcup \varphi_2)]\!]$, it suffices to modify the objective function update formula in Step 6 of Algorithm 1 to,

$$f(s) = f(s) \vee \left(\delta^*(s)^\circ v'(s) \wedge \bigwedge_{s' \in Child(s)} (\delta^*(s) \wedge \delta(s, s'))^\circ v(s')\right).$$

To avoid confusion, we will use $f_\exists$ to denote the computation result of $\exists(\varphi_1 \sqcup \varphi_2)$ and $f_\forall$ to denote the computation result of$\forall(\varphi_1 \sqcup \varphi_2)$.

Theorem 3 guarantees the correctness of Algorithm 1.

The following delineates an algorithm aimed at addressing the model-checking challenge pertinent to the QFCTL formula, specifically $Q_P(\bigcirc\varphi)$.

Algorithm 2, formulated explicitly in alignment with the stipulations of Definition 6 to directly compute $Q_P(\bigcirc\varphi)$, exhibits a correctness that is patently obvious.

Below is presented an algorithm for solving the model checking problem of QFCTL formula $Q_P(\varphi_1 \sqcup \varphi_2)$.

Algorithm 2 The Computation Algorithm of $Q_P(\bigcirc\varphi)$.

Input: A QFCTL formula $Q_P(\bigcirc\varphi)$, a FKS $M = (S, I, \delta, AP, L)$, with the information fusion operator $\circ \in \{\wedge, \cdot, \oplus_\lambda\}$.

Solution process:

```
LOOP $\forall s \in S, I(s) > 0$
    $[\![s, Q_P(\psi)]\!]$=0;
    LOOP $\forall s' \in Child(s), \delta(s, s') > 0$
        $[\![s, Q_P(\psi)]\!] = (I(s) \wedge \delta(s, s')) \circ [\![s', \varphi]\!]$;
        IF $[\![s, Q_P(\psi)]\!] \in P$and $[\![s, Q_P(\psi)]\!]$=0
            $[\![s, Q_P(\psi)]\!]$=1;
        END
        ELSE IF $[\![s, Q_P(\psi)]\!] \notin P$and $[\![s, Q_P(\psi)]\!]$=1
            $[\![s, Q_P(\psi)]\!]$=0.5;
            BREAK;
        END
    END
END
```

Output: $\forall s \in S$ output $[\![s, Q_P(\psi)]\!]$.

Algorithm 3 The Computation Algorithm of $Q_P(\varphi_1 \sqcup \varphi_2)$.

Input: A QFCTL formula $Q_P(\varphi_1 \sqcup \varphi_2)$, an FKS $M = (S, I, \delta, AP, L)$, with the information fusion operator $\circ \in \{\wedge, \cdot, \oplus_\lambda\}$.

Solution process:

```
Algorithm 1 is employed to obtain the fuzzy sets $f_\exists$ and $f_\forall$;
LOOP $\forall s \in S, I(s) > 0$
    IF $P = [\alpha, \beta]$
        IF $f_\forall(s) > \beta$ or $f_\exists(s) < \alpha$ THEN $[\![s, Q_P(\psi)]\!] = 0$;
        IF $\alpha \le f_\forall(s) \le f_\exists(s) \le \beta$ THEN $[\![s, Q_P(\psi)]\!] = 1$;
        IF $f_\forall(s) < \alpha \le f_\exists(s) \le \beta$ or $\alpha \le f_\forall(s) \le \beta < f_\exists(s)$ THEN $[\![s, Q_P(\psi)]\!] = 0.5$;
    END
    IF $P = (\alpha, \beta)$
        IF $f_\forall(s) \ge \beta$ or $f_\exists(s) \le \alpha$ THEN $[\![s, Q_P(\psi)]\!] = 0$;
        IF $\alpha < f_\forall(s) \le f_\exists(s) < \beta$ THEN $[\![s, Q_P(\psi)]\!] = 1$;
        IF $f_\forall(s) \le \alpha < f_\exists(s) < \beta$ or $\alpha < f_\forall(s) < \beta \le f_\exists(s)$ THEN $[\![s, Q_P(\psi)]\!] = 0.5$;
    END
    IF $P = (\alpha, \beta]$
        IF $f_\forall(s) > \beta$ or $f_\exists(s) \le \alpha$ THEN $[\![s, Q_P(\psi)]\!] = 0$;
        IF $\alpha < f_\forall(s) \le f_\exists(s) \le \beta$ THEN $[\![s, Q_P(\psi)]\!] = 1$;
        IF $f_\forall(s) \le \alpha < f_\exists(s) \le \beta$ or $\alpha < f_\forall(s) < \beta < f_\exists(s)$ THEN $[\![s, Q_P(\psi)]\!] = 0.5$;
    END
    IF $P = [\alpha, \beta)$
        IF $f_\forall(s) \ge \beta$ or $f_\exists(s) < \alpha$ THEN $[\![s, Q_P(\psi)]\!] = 0$;
        IF $\alpha \le f_\forall(s) \le f_\exists(s) < \beta$ THEN $[\![s, Q_P(\psi)]\!] = 1$;
        IF $f_\forall(s) < \alpha \le f_\exists(s) < \beta$ or $\alpha \le f_\forall(s) < \beta \le f_\exists(s)$ THEN $[\![s, Q_P(\psi)]\!] = 0.5$;
    END
END
```

Output: $\forall s \in S$ output $[\![s, Q_P(\psi)]\!]$.

Theorem 2 and Corollary 1 guarantee the correctness of Algorithm 3.

Theorem 4. *Let φ be a QFCTL state formula, $M = (S, I, \delta, AP, L)$ be a FKS, and $P \subseteq [0,1]$ be a quality predicate. Then, the time complexity of the model checking problem for φ on M is $O(|\varphi| \cdot |Supp(I)|.(|S| + |Supp(\delta)|))$, and the space complexity is $O(|S| + |Supp(\delta)|)$.*

Proof: The primary computational time is spent on evaluating $\exists(\varphi_1 \sqcup \varphi_2)$ and $\forall(\varphi_1 \sqcup \varphi_2)$ within Algorithm 1. The algorithm consists of two nested loops: the outer loop iterates over $|Supp(I)|$, and the inner loop iterates over $|S|$. Within the inner loop, there is a nested operation of supremum $\bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ v(s')$ (Step 7 of Algorithm 1). Since $\sum_{s \in S} \sum_{s' \in Child(s)} 1 = |Supp(\delta)|$, a total of $3|Supp(\delta)|$ operations are performed (One for each successor state s', with three operations $\wedge, \circ$ and $\vee$ each executed once). Steps 8, 9, and 10 of Algorithm 1 execute a total of $4|S|$ operations (During Each loop iteration performs one minimum operation and three assignment operations). Therefore, the overall time complexity of Algorithm 1 is $O(|Supp(I)|.(|S| + |Supp(\delta)|))$. The space overhead is used to store state and transition function values, which can be reused. Hence, the space complexity is $O(|S| + |Supp(\delta)|)$. Consequently, the time complexity of the model checking problem for φ on M is $O(|\varphi| \cdot |Supp(I)|.(|S| + |Supp(\delta)|))$, and the space complexity is $O(|S| + |Supp(\delta)|)$.□

5. QFCTL* and Its Model Checking

To more precisely characterize the satisfaction of a QFCTL path formula ψ on the set of paths originating from any state s, under the constraint of a quality predicate P, we introduce a quantifier characteristic predicate operator $Q_P^*(.)$ into QFCTL, resulting in an Enhanced Fuzzy Computation Tree Temporal Logic with Quality Constraints, denoted as QFCTL*. Furthermore, we discuss the logical properties and model checking techniques of QFCTL*.

QFCTL* differs from QFCTL by incorporating an additional quantifier characteristic predicate operator $Q_P^*(.)$ and discuss the associated logical properties and model checking algorithms related to it.

Definition 8 [Semantics of the Operator $\boldsymbol{Q_P^*(.)}$] *Let $M = (S, I, \delta, AP, L)$ be an FKS (Fuzzy Kripke Structure), $P \subseteq [0,1]$ be a quality predicate, and ψ be a QFCTL path formula. For all $s \in S$, the semantics of the operator $Q_P^*(.)$ is defined as follows:*

$$[\![s, Q_P^*(\psi)]\!] = \frac{|\{[\![\pi, \psi]\!] | \pi \in Path(s), [\![\pi, \psi]\!] \in P\}|}{|\{[\![\pi, \psi]\!] | \pi \in Path(s)\}|} \tag{5}$$

$Q_P(\psi)$ qualitatively characterizes the satisfaction of the property formula ψ on $Path(s)$ in terms of three scenarios: fully satisfied, fully unsatisfied, or partially satisfied. $Q_P^*(.)$ provides a more granular characterization of the satisfaction of the property formula ψ on $Path(s)$, specifically quantifying the proportion of paths in $Path(s)$ that satisfy the constraint P with respect to ψ.

Example 3 (Continuing from Example 2). *The QFCTL* formula $\varphi_{10} = Q_{>0.6}^*(0.5_{ne}(f) \sqcup e)$ represents the proportion of all treatment plans during the course of treatment that maintain the patient's condition with a necessity of at least "fine" (with a threshold of 0.5) until there is a transition to "excellent" with a possibility greater than 0.6. For instance, if $[\![s_0, Q_{>0.6}^*(0.5_{ne}(f) \sqcup e)]\!] = 0.2$, this signifies that 20% of the treatment plans fulfill the criterion where the patient's condition remains with a necessity of at least "fine" (with a threshold of 0.5) throughout the treatment process until there is a transition to "excellent" with a likelihood exceeding 0.6.*

The following explains that the sets in both the numerator and the denominator of formula (5) are finite sets, indicating that Definition 8 is well-defined.

For all $\pi \in Path(s)$, $n \in \mathbb{N}^+$, denote $\pi(n) = \pi_0, \pi_1, \cdots, \pi_{n-1}$ as the prefix path of π with length n. We define the semantics of QFCTL path formula $\varphi_1 \sqcup \varphi_2$ on $\pi(n)$ as follows:

$$[\![\pi(n), \varphi_1 \sqcup \varphi_2]\!] = \bigvee_{0 \le i < n} \left(\bigwedge_{0 \le j < i} \delta_j^*(\pi) \circ [\![\pi_j, \varphi_1]\!] \wedge \delta_i^*(\pi) \circ [\![\pi_i, \varphi_2]\!] \right). \tag{6}$$

Lemma 1. *Let $M = (S, I, \delta, AP, L)$ be an FKS (Finite Kripke Structure), $P \subseteq [0,1]$ be a quality constraint predicate, and ψ is a QFCTL path formula. Then, for all $s \in S$,*

$$\{[\![\pi, \psi]\!] | \pi \in Path(s)\} = \{[\![\rho(|S|), \psi]\!] | \rho \in Path(s)\}.$$

doi:10.20944/preprints202410.1921.v1

Proof: When ψ is a QFCTL state formula (where a state formula is a special type of path formula with semantics defined at $\pi_0 = s$), the conclusion holds. We consider two cases: $\psi = \bigcirc\varphi$ and $\psi = \varphi_1 \sqcup \varphi_2$.

When $\psi = \bigcirc\varphi$, we have $\llbracket \pi, \bigcirc\varphi \rrbracket = (I(\pi_0) \wedge \delta(\pi_0, \pi_1)) \circ \llbracket \pi_1, \varphi \rrbracket$, where $\llbracket \pi, \bigcirc\varphi \rrbracket$ depends only on π_0 and π_1. Therefore, the conclusion holds.

When $\psi = \varphi_1 \sqcup \varphi_2$, based on Definition 6 and Formula (6), we derive the following conclusion:

$$\llbracket \pi, \varphi_1 \sqcup \varphi_2 \rrbracket = \bigvee_{n \geq |S|} \llbracket \pi(n), \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{7}$$

We prove that for all $n \geq |S|$, there exists a path $\rho \in Path(s)$ such that,

$$\llbracket \pi(n), \varphi_1 \sqcup \varphi_2 \rrbracket \leq \llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{8}$$

When $n = k = |S|$, taking $\rho = \pi$, Inequality (8) holds.

Assume that Equation (8) holds when $n = k \geq |s|$. That is, there exists a path ρ in $Path(s)$ such that:

$$\llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \leq \llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{9}$$

Then, when $n = k + 1$, the classification discussion is as follows:

If there exists a $t \in \mathbb{N}$ such that $0 \leq t < k$ and $\pi_k = \pi_t$, then,

$$\llbracket \pi(k+1), \varphi_1 \sqcup \varphi_2 \rrbracket = \bigvee_{0 \leq i < k+1} (\bigwedge_{0 \leq j < i} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_i^*(\pi) \circ \llbracket \pi_i, \varphi_2 \rrbracket)$$

$$= \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < k} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_k^*(\pi) \circ \llbracket \pi_k, \varphi_2 \rrbracket)$$

$$= \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < k} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_k^*(\pi) \circ \llbracket \pi_t, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < t} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_t^*(\pi) \circ \llbracket \pi_t, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigvee_{0 \leq i < k} (\bigwedge_{0 \leq j < i} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_i^*(\pi) \circ \llbracket \pi_i, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket$$

$$= \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \leq \llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|s|), \varphi_1 \sqcup \varphi_2 \rrbracket \quad \text{(Inductive Hypothesis (9))}$$

If for all $t < k$, we have $\pi_k \neq \pi_t$, then since $k + 1 > |S|$, there must exist $0 < l < h < k$ on the path $\pi(k+1)$ before the state π_k, such that $\pi_l = \pi_h$, forming a loop. By removing the path segment $\pi_{l+1}, \cdots, \pi_{h-1}, \pi_h$ from the path, we construct a new path ρ as follows:

$$\rho = \pi_0, \pi_1, \cdots, \pi_l, \pi_{h+1}, \cdots, \pi_k, \cdots.$$

Numbering sequentially, we obtain:

$$\rho = \rho_0, \rho_1, \cdots, \rho_{k-(h-l)}, \cdots.$$

Where $\rho_0 = \pi_0 = \text{s}$, for all $i \leq l$, $\pi_i = \rho_i$ and for all $i > h$, $\pi_i = \rho_{i-(h-l)}$.

$$\llbracket \pi(k+1), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < k} \delta_j^*(\pi) \circ \llbracket \pi_j, \varphi_1 \rrbracket \wedge \delta_k^*(\pi) \circ \llbracket \pi_k, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < k-(h-l)} \delta_j^*(\rho) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_k^*(\pi) \circ \llbracket \pi_k, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < k-(h-l)} \delta_j^*(\rho) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_{k-(h-l)}^*(\rho) \circ \llbracket \rho_{k-(h-l)}, \varphi_2 \rrbracket)$$

$$\leq \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigvee_{0 \leq i < k} (\bigwedge_{0 \leq j < k} \delta_j^*(\rho) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_k^*(\rho) \circ \llbracket \rho_k, \varphi_2 \rrbracket))$$

$$= \llbracket \pi(k), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(k), \varphi_1 \sqcup \varphi_2 \rrbracket$$

$$\leq \llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \qquad \text{(Inductive Hypothesis (9))}$$

In this way, we have proven Inequality (8).

Furthermore, it is evident that for all $n = k \geq |s|$, $\llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \leq \llbracket \pi(n), \varphi_1 \sqcup \varphi_2 \rrbracket$. Combining this with Inequality (8), we have:

For all $n \geq |S|$, there exists a path $\rho \in Path(s)$, such that ,

$$\llbracket \pi(n), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \pi(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket.$$

Furthermore, combining this with Equation Inequality (7), we obtain:

$$\llbracket \pi, \varphi_1 \sqcup \varphi_2 \rrbracket \in \{ \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket | \rho \in Path(s) \}. \tag{10}$$

For all $s \in S$, let us arbitrarily choose any $\rho(|S|) = \rho_0, \rho_1, \cdots, \rho_{|S|-1}$, $(\rho_0 = s)$. For any $\rho_{|S|} \in Child(\rho_{|S|-1})$, construct $\rho(|S|+1) = \rho_0, \rho_1, \cdots, \rho_{|S|-1}, \rho_{|S|}$. It is evident that $\llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \leq \llbracket \rho(|S|+1), \varphi_1 \sqcup \varphi_2 \rrbracket$. Continuing the discussion as follows:

If there exists a t such that $0 \leq t < |S|$ and $\rho_{|S|} = \rho_t$.

$\llbracket \rho(|S|+1), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < |S|} \delta_j^*(\pi) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_{|S|}^*(\pi) \circ \llbracket \rho_{|s|}, \varphi_2 \rrbracket)$

$\leq \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigwedge_{0 \leq j < t} \delta_j^*(\pi) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_t^*(\pi) \circ \llbracket \rho_t, \varphi_2 \rrbracket)$

$\leq \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \bigvee_{0 \leq i < |S|} (\bigwedge_{0 \leq j < i} \delta_j^*(\pi) \circ \llbracket \rho_j, \varphi_1 \rrbracket \wedge \delta_i^*(\pi) \circ \llbracket \rho_i, \varphi_2 \rrbracket))$

$= \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \vee \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket$.

In this way, we obtain, $\llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \rho(|S|+1), \varphi_1 \sqcup \varphi_2 \rrbracket$. By repeating this process, we can derive that,

$$\llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket = \lim_{n \to \infty} \llbracket \rho(n), \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{11}$$

Denote the limit as $\lim_{n \to \infty} \rho(n) = \pi$, and we obtain that,

$$\llbracket \rho(|s|), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \pi, \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{12}$$

If for all $t < |S|$, it holds that $\rho_{|S|} \neq \rho_t$, then there must exist $0 < l < h < |S|$, such that $\rho_l = \rho_h$, generating a cycle. After traversing the path segment $\pi_{l+1}, \cdots, \pi_{h-1}, \pi_h$ w times, we obtain the path segment:

$\rho(|S| + w(h-l)) = \rho_0, \rho_1, \cdots, \rho_l, (\rho_{l+1}, \cdots, \rho_{h-1}, \pi_h)^w \rho_{h+1}, \cdots, \rho_{|S|}$.

The following conclusion is readily apparent on $\rho(|S| + w(h-l))$.

$$\llbracket \rho(|S| + w(h-l)), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \rho(|s|), \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{13}$$

Denote $\lim_{w \to \infty} \rho(|S| + w(h-l)) = \pi'$. We have proven that,

$$\llbracket \rho(|s|), \varphi_1 \sqcup \varphi_2 \rrbracket = \llbracket \pi', \varphi_1 \sqcup \varphi_2 \rrbracket. \tag{14}$$

From Equations (12) and (14), we obtain that,

$$\llbracket \rho(|S|), \varphi_1 \sqcup \varphi_2 \rrbracket \in \{\llbracket \pi, \varphi_1 \sqcup \varphi_2 \rrbracket | \pi \in Path(s)\}. \tag{15}$$

By synthesizing the conclusions (10) and (15), we can deduce that the conclusion of Lemma 1 holds. □

Theorem 5 [Decision Theorem for the Operator $Q_P^*(.)$] *Let $M = (S, I, \delta, AP, L)$ be an FKS, where $P \subseteq [0,1]$ is a quality predicate, and ψ is a QFCTL path formula. For all $s \in S$, it holds that,*

$$\llbracket s, Q_P^*(\psi) \rrbracket = \frac{|\{\llbracket \pi(|S|), \psi \rrbracket | \pi \in Path(s), \llbracket \pi(|S|), \psi \rrbracket \in P\}|}{|\{\llbracket \pi(|S|), \psi \rrbracket | \pi \in Path(s)\}|} \tag{16}$$

Based on Lemma 1 and Definition 8, Theorem 5 is established. Theorem 5 clarifies that the semantic of the operator $Q_P^*(.)$ is well-defined, and it also furnishes a foundation for model checking the operator $Q_P^*(.)$.

Proposition 3 [The fundamental properties of quantified characteristic predicate operators] *Let $M = (S, I, \delta, AP, L)$ be an FKS, ψ be a QFCTL path formula, and $P \subseteq [0,1]$ be a quality predicate. Let $\bar{P}$ denotes the complement of P in [0,1]. Then,*

(1) $Q_P^*(\psi) + Q_{\bar{P}}^*(\psi) = 1, Q_P^*(\neg\psi) = Q_{\bar{P}}^*(\psi)$;
(2) $Q_P^*(\psi) = 0$ *if and only if* $Q_P(\psi) = 0$;
(3) $Q_P^*(\psi) = 1$ *if and only if* $Q_P(\psi) = 1$.

The proof of Conclusion (1) of Proposition 3 is similar to the proof of Conclusion (1) of Theorem 2. The proofs of Conclusion (2) and Conclusion (3) of Proposition 3 are evident.

Next, we present the model-checking algorithm for $Q_P^*(\psi)$. We only need to provide the algorithm for the case when $\psi = \varphi_1 \sqcup \varphi_2$, as other cases are either trivial and can be directly computed based on Definition 6 and Theorem 5, or they can be reduced to the case of $\psi = \varphi_1 \sqcup \varphi_2$

Algorithm 4 Computing $Q_P^*(\varphi_1 \sqcup \varphi_2)./\psi = \varphi_1 \sqcup \varphi_2$

Input：A QFCTL formula $Q_P^*(\varphi_1 \sqcup \varphi_2)$, an FKS $M = (S, I, \delta, AP, L)$, with the information fusion operator $\circ \in \{\wedge, \cdot, \oplus_\lambda\}$.

Solution process:

```
1 Initialize：Fuzzy sets $v, v': S \longrightarrow [0,1]$, $\forall s \in S$, $v(s) = [\![s, \varphi_2]\!]$, $v'(s) = [\![s, \varphi_1]\!]$.
2 LOOP $\forall s \in S, I(s) > 0$
3     $\delta^*(s) = I(s)$;
4     $f(s) = \delta^*(s)^\circ v(s)$;// Initialize the objective function value for the source state $s$.
5     $Nu(s)= De(s) = \emptyset$;// $Nu(s)$ represents the set in the numerator of Equation (16), while $De(s)$ represents
          the set in the denominator of Equation (16).
6     $Depth(s) = 0$; // $Depth(s)$-count represents the depth of the extension for s.
7     Initialize empty stack OPEN and table TREE //The stack OPEN is utilized for depth-prioritized
      expansion of $M$ originating from node s, whereas the TREE table meticulously archives the resultant
      tree generated from s as its root, ensuing from the expansion of $M$.
8     Push $s$ onto stack OPEN;
9     $s.Father$ =TREE.head; //$s$ is the root node of TREE.
10    $Cn$(TREE. head)=0;// Record the number of child nodes of TREE.head.
11    LOOP stack OPEN is not empty.
12        The top element $\mu$ is popped from stack OPEN.
13        $\mu.Father.Child[Cn(\mu.Father)] = \mu$// Store node $\mu$ in table TREE.
14        $Cn(\mu)$=0;// $Cn(\mu)$ record the number of child nodes of node $\mu$.
15        IF $Depth(\mu) < |S|$// The maximum search depth is $|S| - 1$.
16            LOOP for all $s' \in Child(\mu)$
17                $f(s') = f(\mu) \vee (\delta^*(\mu)^\circ v'(\mu) \wedge (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ v(s'))$;//Update the objective
                          function value for the state $s'$.
18                $\delta^*(s') = \delta^*(\mu) \wedge \delta(\mu, s')$;// Calculate the path reachability from the root node
                          to node $s'$.
19                $\mu \cdot Child[Cn(\mu)] = s'$;// Node $s'$, as a child node of node $\mu$ in the spanning
                      tree, is stored in table TREE.
20                $Cn(\mu) + +$;// The count of $\mu$'s child nodes is incremented by one.
21                $s'.Father = \mu$;// Set the parent node pointer of $s'$ to point to node $\mu$, in
                    preparation for expanding $s'$ in TREE.
22                Push $s'$onto stack OPEN.;
23                $Depth(s') = Depth(\mu) + 1$;
24            END
25        END
26        ELSE IF $Depth(\mu) = |S|$// Reach the deepest level of the search.
27            IF $f(\mu) > 0$ THEN $De(s) = De(s) \cup \{f(\mu)\}$;//A path $\pi(|S|) = s, \cdots, \mu$ is found.
28            IF $f(\mu) \in P$ THEN $Du(s) = Du(s) \cup \{f(\mu)\}$;
              //$f(\mu) \in \{[\![\pi(|S|), \psi]\!] \in P | \pi \in Path(s)\}$.
29        END
39    END
31    $q(s) = |Du(s)|/|De(s)|$;
32 END
```

Output： $\forall s \in S, I(s) > 0$, output $q(s)$.

Theorem 6 *Let $M = (S, I, \delta, AP, L)$ be an FKS, where $P \subseteq [0,1]$ is a quality predicate, and $\psi = \varphi_1 \sqcup \varphi_2$ is a QFCTL path formula. For all $s \in S$, $I(s) > 0$, the output of Algorithm 4 ensures $q(s) = \left[\!\!\left[s, Q_P^*(\varphi_1 \sqcup \varphi_2)\right]\!\!\right]$. The time complexity of model checking $Q_P^*(\varphi_1 \sqcup \varphi_2)$ is $O(suup(I) \cdot |S|^{|S|-1})$, and the space complexity is $O(|S|^{|S|-1})$.*

Proof: Algorithm 4 employs a bounded depth-first search algorithm with a maximum search depth of $|S| - 1$. Starting from each initial state s, it produces a specific path that begins at s with $Depth(s) = 0$ and terminates at a state μ with $Depth(\mu) = |S| - 1$, as detailed below,

$$\pi(|S|) = s, s', \cdots, \mu', \mu.$$

During the iterative search process, the objective function $f(s)$ is continuously evaluated (as specified in step 17 of Algorithm 4), making it straightforward to obtain the following results,

doi:10.20944/preprints202410.1921.v1

$f(s) = I(s)^{\circ}v(s) \vee (\delta^*(s)^{\circ}v'(s) \wedge (\delta^*(s) \wedge \delta(\mu, s'))^{\circ}v(s')) \vee \cdots$

$\vee (\delta^*(\mu')^{\circ}v'(\mu') \wedge (\delta^*(\mu') \wedge \delta(\mu', \mu))^{\circ}v(\mu))$

$= \bigvee_{0 \le i < |S|} (\bigwedge_{0 \le j < i} \delta_j^*(\pi) \circ [\![\pi_j, \varphi_1]\!] \wedge \delta_i^*(\pi) \circ [\![\pi_i, \varphi_2]\!])$

$= [\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!]$.

Step 27: If $[\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!] = f(s) > 0$, then $De(s) = De(s) \cup \{f(\mu)\}$;

Step 28: If $[\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!] = f(s) \in P$, then $Du(s) = Du(s) \cup \{f(\mu)\}$.

Consequently, it is straightforward to derive the following:

$De(s) = \{[\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!] | \pi \in Path(s)\}$;

$Du(s) = \{[\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!] | \pi \in Path(s), [\![\pi(|S|), \varphi_1 \sqcup \varphi_2]\!] \in P\}$.

Therefore, $q(s) = [\![s, Q_P^*(\varphi_1 \sqcup \varphi_2)]\!]$. The correctness of Algorithm 4 is thereby established.

During the search process, for each node generated, the seven main computations from Step 17 to Step 23 are executed, and the resulting nodes are continuously stored in the generated tree, TREE. The primary time and space costs associated with generating TREE are both linearly related to the number of nodes in TREE. Specifically, TREE has 1 node at level 0, at most $|S|$ nodes at level 1, at most $|S|^2$ nodes at level 2, ..., and at most $|S|^{|S|-1}$ nodes at the deepest level $|S| - 1$. Therefore, the number of nodes in TREE is given by $(|S|^{|S|} - 1)/(|S| - 1)$. The number of elements in the sets $De(s)$ and $Du(s)$ does not exceed the number of leaf nodes in TREE, which is $|S|^{|S|-1}$. Consequently, the time complexity for computing $De(s)$ and $Du(s)$ is $O(|S|^{|S|-1})$. Since this search process needs to be repeated for all $s \in S, I(s) > 0$, the overall time complexity becomes $O(suup(I) \cdot |S|^{|S|-1})$. However, the space occupied by TREE, $De(s)$ and $Du(s)$ can be reused, so the space complexity is determined by the size of TREE, which is $O(\left|S\right|^{|S|-1})$. □

Theorem 7 *Let* $M = (S, I, \delta, AP, L)$ *be an FKS,* φ *be a QFCTL* state formula, and* $P \subseteq [0,1]$ *be a quality predicate. The time complexity of QFCTL* model checking is* $O(|\varphi| \cdot suup(I) \cdot |S|^{|S|-1})$, *and the space complexity is* $O(|S|^{|S|-1})$.

Based on Theorem 6, when considering an arbitrary QFCTL* formula φ, the complexity of model checking is the product of the length of φ (denoted as $|\varphi|$) and the complexity of Algorithm 4. Therefore, the time complexity of QFCTL* model checking is $O\left(|\varphi| \cdot suup(I) \cdot |S|^{|S|}\right)$. The space complexity remains $O(\left|S\right|^{|S|-1})$ due to the reusable nature of the space.

6. Illustrative Examples

Up to this point, we have expounded on the theoretical underpinnings of QFCTL model checking. In this section, we will elucidate the modeling advantages of QFCTL, and the efficacy of our model checking techniques through a series of illustrative examples. All the model-checking examples provided herein are executed within the FKS depicted in Figure 1.

First, we present the outcomes of evaluating QFCTL formulas φ_1 through φ_6 from Example 2 using Algorithm 1. The information fusion operator "$\circ$" is instantiated as "$\oplus_{0.3}$".

The computation of $[\![s_0, \varphi_1]\!] = \left[\!\left[s_0, \exists\Diamond(0.8_{cf}(e))\right]\!\right]$ requires $|S| = 3$ iterations, where s_0 denotes the initial state, and $|S|$ indicates the cardinality of the state space S.

The initial value is given by $f(s_0) = I(s_0) \oplus_{0.3} s_0\left(0.8_{cf}(e)\right) = 0.3 + 0 = 0.3$. Furthermore, $\delta^*(s_0) = I(s_0) = 1$.

Let $\mu = s_0$; the first iteration proceeds as follows:

$$f(\mu) = f(\mu) \vee \bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^{\circ}s'(0.8_{cf}(e)) \qquad (17)$$

$= 0.3 \vee ((I(s_0) \wedge \delta(s_0, s_0)) \oplus_{0.3} s_0(0.8_{cf}(e)))$

$\vee ((I(s_0) \wedge \delta(s_0, s_1)) \oplus_{0.3} s_1(0.8_{cf}(e)) \vee ((I(s_0) \wedge \delta(s_0, s_2)) \oplus_{0.3} s_2(0.8_{cf}(e)))$

$= 0.3 \vee (0.3 \oplus_{0.3} 0) \vee (0.8 \oplus_{0.3} 0.8 \times 0.5) \vee (0.2 \oplus_{0.3} 0.8 \times 1) = 0.62$.

$\delta^*(\mu) = \delta^*(\mu) \wedge \delta(\mu, s_1) = 0.8, f(s) = f(\mu) = 0.62, \mu = s_1$.

The second iteration proceeds as follows:

$$f(\mu) = f(\mu) \vee \bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(0.8_{cf}(e))$$

$$= 0.62 \vee (0.8 \wedge \delta(s_1, s_0)) \oplus_{0.3} s_0(0.8_{cf}(e)))$$

$$\vee ((0.8 \wedge \delta(s_1, s_1)) \oplus_{0.3} s_1(0.8_{cf}(e)) \vee (0.8 \wedge \delta(s_1, s_2)) \oplus_{0.3} s_2(0.8_{cf}(e)))$$

$$= 0.62 \vee (0.4 \oplus_{0.3} 0) \vee (0.7 \oplus_{0.3} 0.8 \times 0.5) \vee (0.7 \oplus_{0.3} 0.8 \times 1) = 0.77.$$

$$\delta^*(\mu) = \delta^*(\mu) \wedge \delta(\mu, s_2) = 0.7, f(s) = f(\mu) = 0.77, \mu = s_2.$$

The third iteration proceeds as follows:

$$f(\mu) = f(\mu) \vee \bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(0.8_{cf}(e))$$

$$= 0.77 \vee (0.7 \wedge \delta(s_2, s_0)) \oplus_{0.3} s_0(0.8_{cf}(e)))$$

$$\vee ((0.7 \wedge \delta(s_2, s_1)) \oplus_{0.3} s_1(0.8_{cf}(e)) \vee (0.7 \wedge \delta(s_2, s_2)) \oplus_{0.3} s_2(0.8_{cf}(e)))$$

$$= 0.62 \vee (0 \oplus_{0.3} 0) \vee (0.5 \oplus_{0.3} 0.8 \times 0.5) \vee (0.7 \oplus_{0.3} 0.8 \times 1) = 0.77.$$

$$\delta^*(\mu) = \delta^*(\mu) \wedge \delta(\mu, s_2) = 0.7, f(s) = f(\mu) = 0.77, \mu = s_2.$$

Ultimately, the evaluation yields, $[\![s_0, \varphi_1]\!] = f(\mu) = 0.77$.

When computing $[\![s_0, \varphi_2]\!] = [\![s_0, \forall\Diamond(0.8_{cf}(e))]\!]$, one merely needs to substitute Formula (17) with the following Formula (18).

$$f(\mu) = f(\mu) \wedge \bigvee_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(0.8_{cf}(e)). \tag{18}$$

Following three iterations, the computed value is $[\![s_0, \varphi_2]\!] = 0.3$.

When computing $[\![s_0, \varphi_3]\!] = [\![s_0, \exists\Box(f \oplus_{0.4} e)]\!]$, where $f(s_0) = I(s_0) \oplus_{0.3} s_0(f \oplus_{0.4} e)$, one merely needs to substitute Formula (17) with the following Formula (19).

$$f(\mu) = f(\mu) \vee \bigwedge_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(f \oplus_{0.4} e). \tag{19}$$

Upon completing another trio of iterations, the calculated outcome is $[\![s_0, \varphi_3]\!] = 0.328$.

In computing $[\![s_0, \varphi_4]\!] = [\![s_0, \forall\Box(f \oplus_{0.4} e)]\!]$, where $f(s_0) = I(s_0) \oplus_{0.3} s_0(f \oplus_{0.4} e)$, replace Formula (17) with the following Formula (20).

$$f(\mu) = f(\mu) \vee \bigwedge_{s' \in Child(\mu)} (\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(f \oplus_{0.4} e). \tag{20}$$

After undergoing three iterations, the solution obtained is $[\![s_0, \varphi_4]\!] = 0.088$.

Upon completing another trio of iterations, the calculated outcome is $[\![s_0, \varphi_3]\!] = 0.328$.

In computing $[\![s_0, \varphi_5]\!] = [\![s_0, \exists(0.5_{ne}(f) \sqcup e)]\!]$, where $f(s_0) = I(s_0) \oplus_{0.3} s_0(e)$, replace Formula (17) with the following Formula (21).

$$f(\mu) = f(\mu) \vee \bigwedge_{s' \in Child(\mu)} ((\delta^*(\mu)^\circ \mu(0.5_{ne}(f))) \wedge ((\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(e))). \tag{21}$$

After undergoing three iterations, the solution obtained is $[\![s_0, \varphi_5]\!] = 0.685$.

In computing $[\![s_0, \varphi_6]\!] = [\![s_0, \forall(0.5_{ne}(f) \sqcup e)]\!]$, where $f(s_0) = I(s_0) \oplus_{0.3} s_0(e)$, replace Formula (17) with the following Formula (22).

$$f(\mu) = f(\mu) \wedge \bigwedge_{s' \in Child(\mu)} ((\delta^*(\mu)^\circ \mu(0.5_{ne}(f))) \wedge ((\delta^*(\mu) \wedge \delta(\mu, s'))^\circ s'(e))). \tag{22}$$

After undergoing three iterations, the solution obtained is $[\![s_0, \varphi_6]\!] = 0.3$.

We tabulate the model checking outcomes for properties φ_1 to φ_6 in Model M as follow.

Table 1. Model Checking Results for Properties $\boldsymbol{\varphi_1}$ to $\boldsymbol{\varphi_6}$ in Model M.

$\boldsymbol{\varphi_i}$	$\boldsymbol{\exists\Diamond(0.8_{cf}(e))}$	$\boldsymbol{\forall\Diamond(0.8_{cf}(e))}$	$\boldsymbol{\exists\Box(f \oplus_{0.4} e)}$	$\boldsymbol{\forall\Box(f \oplus_{0.4} e)}$	$\boldsymbol{\exists(0.5_{ne}(f) \sqcup e)}$	$\boldsymbol{\forall(0.5_{ne}(f) \sqcup e)}$
$[\![s_0, \varphi_i]\!]$	0.77	0.3	0.328	0.088	0.685	0.3

In the literature [20,21], PoCTL is studied, which solely utilizes the classical max–min operation for information synthesis, excluding quality constraint operators such as $\lambda_{cf}(\cdot)$, $\lambda_{ne}(\cdot)$, and $\oplus_\lambda$. As a result, it cannot express the more nuanced properties like φ_1 to φ_6 in QFCTL. We have computed the satisfaction values of the PoCTL formulas $\phi_1 = \exists\Diamond e$, $\phi_2 = \forall\Diamond e$, $\phi_3 = \exists\Box(f \wedge e)$, $\phi_3 = \exists\Box(f \wedge$

e), $\phi_4 = \forall\Box(f \wedge e)$, $\phi_5 = \exists(f \sqcup e)$ and $\phi_6 = \exists(f \sqcup e)$, which correspond to these six QFLTL formulas, on the KFS in the example regarding the patient treatment process. The calculation results are presented in Table 2 to facilitate comparison and analysis between QFCTL and PoCTL.

Table 2. Model Checking Results for PoCTL formulas $\boldsymbol{\phi_1}$ to $\boldsymbol{\phi_6}$ in Model M.

$\boldsymbol{\phi_i}$	$\exists\Diamond\boldsymbol{e}$	$\forall\Diamond\boldsymbol{e}$	$\exists\Box(\boldsymbol{f} \wedge \boldsymbol{e})$	$\forall\Box(\boldsymbol{f} \wedge \boldsymbol{e})$	$\exists(\boldsymbol{f} \sqcup \boldsymbol{e})$	$\forall(\boldsymbol{f} \sqcup \boldsymbol{e})$
$[\![s_0, \varphi_i]\!]$	**0.7**	0	0	0	**0.1**	0

By conducting a comparative analysis of Tables 1 and 2, we arrive at the following conclusions:

(1) QFCTL demonstrates superior expressivity in comparison to PoCTL. PoCTL is unable to articulate temporal properties that incorporate qualitative constraints, such as φ_1 to φ_6, which specify concrete quality constraints. QFCTL enables a more precise and quantitative portrayal of system attributes, as exemplified by the second row of the two tables, showcasing the diversity of satisfaction values achieved by the corresponding formulas within the system. Evidently, under identical systems, QFCTL formulas differentiate values with greater granularity, whereas PoCTL is relatively coarser.

(2) PoCTL may lead to the loss of information from either side of the path reachability and property formula satisfaction levels. However, QFCTL avoids this deficiency. PoCTL simply employs the "∧" operator to amalgamate the values of path reachability and property formulas, ultimately opting for the lesser value as the satisfaction gauge. For instance, in Table 2, the bold figures represent the path reachability, whereas the non-bold values correspond to property formula values. Conversely, QFCTL in Table 1 perpetually integrates information from both domains, thereby overcoming the information loss in PoCTL.

(3) PoCTL is incapable of distinguishing between the relative importance of sub-formulas within a property formula, nor does it differentiate between the significance of the property formula itself and path reachability. On the other hand, QFCTL adroitly discriminates between such information types. For example, both the PoCTL formulas ϕ_3 and ϕ_4, and the QFCTL formulas φ_3 and φ_4, require simultaneous occurrences of f and e along the path. However, ϕ_3 and ϕ_4 select the lesser value of f and e, whereas φ_3 and φ_4 unify f and e with a weight ratio of 0.4:0.6, signifying the greater importance attributed to e. In the fusion of path reachability and property satisfaction, PoCTL solely picks the minimal value, whereas QFCTL formulas can blend the two using the $\bigoplus_{0.3}$ operation, applying a weight ratio of 0.3:0.7, indicating QFCTL places greater emphasis on property formula satisfaction over path reachability.

(4) PoCTL might induce asynchrony between property formula values and path-reachability information, whereas QFCTL invariably ensures these two aspects of information remain synchronized. As seen in Table 2, the truth values of ϕ_3 and ϕ_4 are always "0," resulting from computations within M where, irrespective of the path taken from s_0, the value of the property formula $f \wedge e$ is always "0." Therefore, regardless of the current path reachability level, taking the minimum with the property formula satisfaction value of "0" results in "0." This clearly instigates a mismatch between path reachability and property formula satisfaction values. The valuation processes of QFCTL formulas φ_3 and φ_4, as evident from Table 1 and the examples, illustrate the remediation of information asynchrony in QFCTL.

Proceeding forth, Algorithm 2 will be applied to conduct model checking for temporal logic properties φ_7 through φ_9.

The temporal logic property is articulated as $\varphi_7 = Q_{\geq 0.3}(0.8_{cf}(e))$, while the quality predicate P_7 is delineated within the interval $[0.3,1)$.

Having previously determined via Algorithm 1 that $[\![s_0, \forall\Diamond(0.8_{cf}(e))]\!] = 0.3$ and $[\![s_0, \exists\Diamond(0.8_{cf}(e))]\!] = 0.77$, it follows that,

$$0.3 \leq \left[\!\!\left[s_0, \forall\Diamond\left(0.8_{cf}(e)\right)\right]\!\!\right] \leq \left[\!\!\left[s_0, \exists\Diamond\left(0.8_{cf}(e)\right)\right]\!\!\right] < 1.$$

Consequently, $[\![s_0, Q_{\geq 0.3}(0.8_{cf}(e))]\!] = 1$.

This result elucidates that across all therapeutic pathways, the criterion which stipulates "the patient's physical condition being maximally in an '*excellent*' state with a possibility of 0.8 occurs with

doi:10.20944/preprints202410.1921.v1

a likelihood of no less than 0.3" is fully satisfied. This affirms that our treatment protocols adhere to the possibility standards for achieving a high level of clinical efficacy.

The temporal logic property $\varphi_8 = Q_{>0.5}(f \oplus_{0.4} e)$, with the quality predicate P_8 delineated as (0.5,1).

Owing to the fact that $[\![s_0, \exists\Box(f \oplus_{0.4} e)]\!] = 0.328 \leq 0.5$, it logically follows that $[\![s_0, Q_{>0.5}(f \oplus_{0.4} e)]\!] = 0$.

This finding elucidates that across every conceivable therapeutic trajectory, the proposition stating "the patient's health status continuously meets the criteria of being '*fine*' with a 40% stipulation and '*excellent*' with a 60% stipulation with a likelihood exceeding 0.5" is not upheld. Consequently, it is affirmed that our current therapeutic paths do not fulfill the possibility threshold for concurrently maintaining these health standards at the specified frequencies.

The temporal logic property $\varphi_9 = Q_{>0.6}(0.5_{ne}(f) \sqcup e)$, complemented by the quality predicate P_9 defined as (0.6,1).

Previously determined, we have $[\![s_0, \forall(0.5_{ne}(f) \sqcup e)]\!] = 0.3$ and,

$[\![s_0, \exists(0.5_{ne}(f) \sqcup e)]\!] = 0.685$.

Given that $[\![s_0, \forall(0.5_{ne}(f) \sqcup e)]\!] \leq 0.6 < [\![s_0, \exists(0.5_{ne}(f) \sqcup e)]\!] < 1$, it is deduced that $[\![s_0, Q_{>0.6}(0.5_{ne}(f) \sqcup e)]\!] = 0.5$.

This reveals that some, but not all, therapeutic courses satisfy the property that "the patient's health status, being maximally under a necessity of 0.5 to be '*fine*', until transitioning to '*excellent*', does so with a possibility greater than 0.6". This indicates a mixed fulfillment of the possibility condition across different treatment pathways.

Proceeding with Algorithm 4, we conduct model checking on the QFCTL* formula $\varphi_{10} = Q^*_{>0.6}(0.5_{ne}(f) \sqcup e)$. The generation tree TREE, produced by the FKS M with the unique initial state s_0 as the root node, as depicted in Figure 7 corresponding to what was shown in Figure 1, illustrates the values calculated for the objective function $f(\mu)$ upon the creation of node μ, which are placed to the right or below each node μ in the tree diagram.

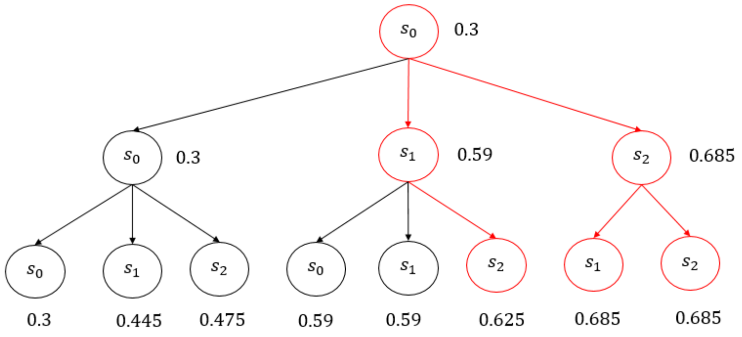

Figure 7. The TREE generated by M in Figure 1.

In TREE, there exist 8 paths of length $|S| = 3$, among which 3 highlighted in red satisfy the following conclusions:

$[\![\pi(3), 0.5_{ne}(f) \sqcup e]\!] \in P_{10} = (0.6,1)$;

$\{[\![\pi(3), 0.5_{ne}(f) \sqcup e]\!] | \pi \in Path(s_0)\} = Du(s_0)$;

$= \{0.3, 0.445, 0.475, 0.59, 0.625, 0.685, 0.685\}$;

$\{[\![\pi(3), 0.5_{ne}(f) \sqcup e]\!] | \pi \in Path(s_0), [\![\pi(3), 0.5_{ne}(f) \sqcup e]\!] \in P_{10}\} = De(s_0)$

$= \{0.625, 0.685, 0.685\}$.

$[\![s_0, Q^*_{>0.6}(0.5_{ne}(f) \sqcup e)]\!] = q(s_0) = |De(s_0)|/|Du(s_0)| = 3/8 = 0.375$.

It indicates that 37.5% of the treatment regimens meet the criterion of the patient's condition being maximally at 'fine' with a necessity of 0.5 throughout the treatment course, until transitioning to 'excellent' with a likelihood greater than 0.6.

These examples illustrate that PoCTL lacks formulas akin to the qualitative QFCTL formulas φ_7 and φ_8, which describe the qualitative fulfillment status of property formulas. Similarly, it does not possess formulas like the quantitative QFCTL* formula φ_9, capable of delineating the quantitative satisfaction condition of property formulas. This unequivocally demonstrates that QFCTL and QFCTL* exhibit superior expressive power over PoCTL, enabling a more refined formal description of temporal system properties. This enhanced capability facilitates a deeper analysis and a more detailed modeling of system temporal behaviors.

These model-checking instances, first and foremost, substantiate the rich expressiveness and practical applicability of QFCTL and QFCTL* proposed in the article, capable of elaborately and accurately depicting the temporal properties of uncertain systems. Secondly, these examples affirm the effectiveness of the model-checking algorithm put forward, enabling automated and efficient formal verification of uncertain systems.

7. Conclusions and Future Work

This paper introduces quality constraint functions and quality predicates into Possibility Computation Tree Logic (PoCTL), proposing Fuzzy Computation Tree Logic with quality Constraints (QFCTL). We explore the logical characteristics of QFCTL and present model checking algorithms for QFCTL on Fuzzy Kripke Structure Systems (FKS). Key contributions include:

(1) Section 2 elaborates on QFCTL syntax in section 2.1, defines its semantics in 2.2, and demonstrates QFCTL's expressive power and practical applicability through illustrative examples in 2.3.

(2) In section 3, we discuss QFCTL's logical features, encompassing equivalences of QFCTL formulas (Theorem 1), partial order relations (Proposition 2), and a complete set of functional operators for QFCTL (Proposition 1). Special focus is given to the basic properties and determination of the Quality Predicate Operator $Q_P(\cdot)$ (Theorem 2, Corollary 1), which signifies formula satisfaction under the constraints of predicate P on relevant paths, yielding truth values of either "1" for full satisfaction, "0" for none, or "0.5" indicating partial satisfaction.

(3) Section 4 delineates a model checking algorithm for QFCTL on FKS, reducing the computation of QFCTL formulas from infinite to finite paths. Model-checking algorithms (Algorithm 1, 2, and 3) are provided with a time complexity proportional to the product of the formula's length and the scale of the FKS ($O(|\varphi|\cdot|Supp(I)|.(|S|+|Supp(\delta)|))$), and space complexity dependent on FKS size ($O(|S|+|Supp(\delta)|)$). Rigorous proofs of the algorithm's validity (Theorem 3) and complexity analysis (Theorem 4) are provided.

(4) Section 5 advances by introducing a quantitative quality predicate operator $Q_P^*(.)$ (Definition 8), characterizing property formula $Q_P^*(\psi)$ satisfaction under predicate P based on the ratio of satisfied paths among those fulfilling ψ. The well-definedness of operator $Q_P^*(.)$ is proven (Theorem 5). Logical traits of QFCTL* are explored alongside the relationship between operators $Q_P(\cdot)$ and $Q_P^*(.)$ (Proposition 3). An algorithm for model checking QFCTL* (Algorithm 4) is presented, accompanied by complexity analysis (Theorem 11).

(5) Section 6 provides ten exemplary QFCTL and QFCTL* formulas($\varphi_1 \sim \varphi_{10}$) applied in model checking scenarios involving patient treatment processes on FKS, as illustrated in literature [24]. These cases attest to the robust expressiveness and practical utility of QFCTL and QFCTL*. They also affirm the automation and effectiveness of our proposed model checking algorithms.

Subsequent primary research efforts will encompass:

(1) The complexity of the QFCTL* model checking algorithm is exponential. We anticipate reducing the complexity of the QFCTL* model checking algorithm through formal reduction methods such as possibility bisimulation and partial-order reduction.

(2) Integrating fuzzy temporal constraints [22,25], path reachability information, and property satisfaction values in a biased, synchronized manner. We aim to investigate fuzzy temporal logic with dual constraints of fuzzy time and quality attributes, thereby enhancing the completeness and accuracy of information representation.

(3) The search and decision problems within QFCTL based on possible decision processes [25,26] remain a domain requiring further exploration. This endeavor will focus on advancing methodologies to address these issues, contributing to the field's mathematical, academic, and readability standards.

These research directions not only aim to tackle existing challenges but also to pave the way for future advancements in the field of fuzzy temporal logic and model checking algorithms.

Author Contributions: X.Y. designed the syntax and semantics of QFCTL, articulated its model checking problem, and provided relevant solution algorithms. S.G. did likewise for QFCTL*, crafting its syntactic and semantic framework, delineating its model checking issue, and offering corresponding algorithmic solutions. Y.L. devoted efforts to investigating the logical properties of both QFLTL and QFCTL. H.L. furnished illustrative examples showcasing QFLTL applications and conducted a meticulous review of the entire paper to guarantee compliance with requisite writing standards, ensuring mathematical rigor, academic integrity, and readability throughout.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No: 12071271,11671244, 12471437), Shaanxi Fundamental Science Research Project for Mathematics and Physics (Grant No: 23JSZ011, 23JSY048), National Key R&Dplan (Grant No: 2020YFC1523305), Key R&Dand transformation plan of Qinghai Province (Grant No: 2022-QY-203), Scientific and Technological Research Fund of Shangluo University (Grant NO. 20SKY021),and Shangluo University Key Disciplines Project, Discipline name：Mathematics.

Data Availability Statement: Data are contained within the article.

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

References

1. Baier C, Katoen J P. Principles of Model Checking. Cambridge: MIT Press, 2008.
2. Edmund M, Grumberg O, Peled D. Model Checking. Cambridge: MIT Press, 1999.
3. P. Camurati and P. Prinetto. Formal verification of hardware correctness: Introduction and survey of current research. *Computer*,**1998**, 21 (7):8–19.
4. Araujo C, Cavalcante E, Batista T, et al. A research landscape on formal verification of software architecture descriptions. *IEEE Access*, **2019**, 7: 171752-171764.
5. Kern C, Greenstreet M R. Formal verification in hardware design: a survey. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, **1999**, 4(2): 123-193.
6. Šimoňák S. Verification of communication protocols based on formal methods integration. *Acta Polytechnica Hungarica*, **2012**, 9(4): 117-128.
7. Boigelot B, Godefroid P. Symbolic verification of communication protocols with infinite state spaces using QDDs. *Formal Methods in System Design, 1999, 14: 237-255.*
8. Avalle M, Pironti A, Sisto R. Formal verification of security protocol implementations: a survey. *Formal Aspects of Computing*, **2014**, 26: 99-123.
9. Coffey T, Dojen R, Flanagan T. Formal verification: an imperative step in the design of security protocols. *Computer Networks, **2003**, 43(5): 601-618.*
10. Zheng Y, Wang L. Consensus of switched multi-agent systems. *IEEE Trans Circ Syst II*, **2016**, 63: 314–318.
11. Zheng Y, Wang L. A novel group consensus protocol for heterogeneous multi-agent systems. *Int J Contr*, **2015**, 106:1–13
12. Li T, Zhang J F. Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises. *IEEE Trans Automat Contr*, **2010**, 55: 2043–2057
13. Li T, Fu M, Xie L, et al. Distributed consensus with limited communication data rate. *IEEE Trans Automat Contr*,**2011**, 56: 279–292.
14. Baier C, Kwiatkowska M. Model checking for a probabilistic branching time logic with fairness. *Distrib Comput*, 1998,11: 125–155.
15. Hart S, Sharir M. Termination of probabilistic concurrent programs. *ACM Trans Prog Lang Syst*, **1983**, 5: 356–380
16. Sultan K, Bentahar J, Wei W, et al. Modeling and verifying probabilistic multi-agent systems using knowledge andsocial commitments. *Expert Syst Appl*, **2014**, 41: 6291–6304
17. Sultan K, Bentahar J, EI-Menshawy M. Model checking probabilistic social commitments for intelligent agent com-munication. *Appl Softw Comput*, **2014**, 22: 397–409
18. Chechik M, Devereux B, Easterbrook S, et al. Multi-valued symbolic model-checking. *ACM Trans Softw Eng Method*,2003, 12: 371–408.
19. Chechik M, Gurfinkel A, Devereux B, et al. Data structures for symbolic multi-valued model-checking. *Formal MethodsSyst Des*, **2006**, 29: 295–344.
20. Li Y M, Li Y N, Ma Z Y. Computation tree logic model checking based on possibility measures. *Fuzzy Sets and Systems*, **2015**, 262, 44-59.
21. Li Y M, Ma Z. Quantitative computation tree logic model checking based on generalized possibility measures. *IEEE Transactions on Fuzzy Systems*, **2015**, 23(6), 2034-2047.
22. A.Frigeri, L.Pasquale, P.Spoletini. Fuzzy time in linear temporal logic. *ACM Transaction on Computational Logic*, **2014,**15(4): 30-53.
23. Almagor S, Boker U, Kupferman O. Formally reasoning about quality. *Journal of the ACM*, 2016, 63(3): 1-56.

24. Yu, X.; Li, Y.; Geng, S. Fuzzy Linear Temporal Logic with Quality Constraints. *Mathematics* **2024**, 12, 3148. https://doi.org/10.3390/math12193148.
25. Li, Y.; Liu, W.; Wang, J.; Yu, X.; Li, C. Model checking of possibilistic linear-time properties based on generalized possibilistic decision processes. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 3495–3506.
26. Liu WN, Li Y M. Optimal strategy model checking in possibilistic decision processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023, (53)10: 6620–6632.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.