mathematics

MDPI

Article

Fuzzy Linear Temporal Logic with Quality Constraints

Xianfeng Yu [1,2], Yongming Li [1,3,*] and Shengling Geng [1]

[1] College of Computer Science, Qinghai Normal University, Xining 810008, China; pioneer.369@163.com (X.Y.)
[2] School of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China
[3] School of Mathematics and Statistics, Shanxi Normal University, Xi'an 710062, China
* Correspondence: liyongm@snnu.edu.cn

Abstract: As an extension of quantitative temporal logic, uncertain temporal logic essentially describes the temporal behavior of uncertain and incomplete systems, thus better solving search and decision-making problems in such systems. Fuzzy linear temporal logic (FLTL) is a focal point in uncertain temporal logic research. However, there are evident shortcomings in the current research outcomes. First, in previous FLTL studies, the practice of obtaining path reachability and formula satisfaction values independently and subsequently selecting the smaller of the two as the satisfaction value metric led to information loss. Furthermore, this simplistic information fusion approach fails to reflect the varying importance of these two types of information to the requirements. Second, computing path reachability and temporal logic formula satisfaction values separately may result in a mismatch between the two pieces of information with respect to the same path segment. Thus, the primary challenge lies in accurately integrating the satisfaction values of temporal logic formulas with the path reachability of the segments that yields these satisfaction values, utilizing various reasonable information synthesis methods, to ensure synchronization between path reachability and formula satisfaction values without incurring information loss. Additionally, it is crucial to reflect the different preference requirements for these two types of information. Moreover, the temporal logic formula characterizes system properties, with its sub-formulas delineating distinct sub-properties. Consequently, considering the varying importance preferences of sub-formulas is also significant. To address these deficiencies, we introduced quality constraint operators into FLTL, resulting in quality-constrained fuzzy linear temporal logic (QFLTL). This incorporation enables the synchronization and comprehensive fusion of path-reachability information and formula satisfaction values within the final semantic metric, thereby resolving the issues related to information synchronization and loss. Furthermore, it can accommodate the differing preference requirements between the two types of information and sub-properties during the information synthesis process. We defined the syntax and semantics of QFLTL and examined its expressive power and properties. Notably, we investigated the decidability of logical decision problems in QFLTL, encompassing validity, satisfiability, and model-checking issues. We proposed corresponding solution algorithms and analyzed their complexities.

Keywords: quality constraints; fuzzy linear temporal logic (FLTL); fuzzy linear temporal logic with quality constraints (QFLTL); model checking; search and decision problems

MSC: 68T37

Citation: Yu, X.; Li, Y.; Geng, S. Fuzzy Linear Temporal Logic with Quality Constraints. *Mathematics* **2024**, *12*, 3148. https://doi.org/10.3390/math12193148

Academic Editor: Michael Voskoglou

Received: 24 May 2024
Revised: 24 August 2024
Accepted: 4 October 2024
Published: 8 October 2024

1. Introduction

Temporal logic is a valuable tool for describing system behaviors over time and finds wide applications in searching and decision-making. Classic temporal logic formulas, such as LTL, CTL, and CTL*, can only characterize the qualitative properties of a system. The expressive power of quantitative temporal logic is more powerful, including probabilistic temporal logic [1–6], possibilistic temporal logic [7,8], and lattice-valued temporal logic [9–12].

The work presented in [7] does not integrate information regarding property satisfaction value and path reachability when evaluating the truth of fuzzy linear temporal logic (FLTL) formulas. Typically, this is achieved by considering the maximum measure of the paths within the path set, where the formula satisfies the predicate constraints. In contrast, in reference [8], the "$\wedge$" (min) operation is utilized for the composite operation involving both the formula satisfaction value and the path-reachability measure. Under the "$\wedge$" operation framework, the composite manipulation of the transition matrix can efficiently compute the quantitative reachable semantics of the possibilistic temporal logic formulas, thereby resolving the corresponding model-checking problem. However, the "$\wedge$" operation selects the smaller value between path-reachability metrics and formula satisfaction values for attributes along the path, potentially leading to information loss. Furthermore, this simplistic approach to information fusion fails to differentiate between the significance of path reachability and attribute formula satisfaction values. In light of this, we propose incorporating a novel fuzzy synthesis operation for information fusion, aimed at overcoming the limitation of information loss inherent in the "$\wedge$" operation. To achieve this, we introduce quality constraint operators that weigh and integrate these two types of information, thereby reflecting the varying levels of importance attributed to each type of information, as per the preference requirements. Additionally, employing quality-constrained operators to constrain sub-property formulas results in a QFLTL formula that is capable of distinguishing the importance of different sub-properties within the resulting temporal logic formula.

On the other hand, when computing the satisfaction value of fuzzy temporal logic (FTL) formulas, as in [8], the semantics of any FTL formula φ are compounded by both the formula satisfaction value and the reachability measure of the entire infinite path. However, when the satisfaction value of the formula is only obtained on a finite fragment of this infinite path, the formula satisfaction value and path-reachability measure are not synchronized. For instance, if formula "$\bigcirc\varphi$" (where "$\bigcirc$" denotes the next time operator) contains only propositional logical operators within φ, it is not necessary to consider the possibility measure of the entire infinite path. Even though the semantics of operators such as "$\Diamond$" (eventually), "$\Box$" (always), and "$\bigcup$" (until) inherently encompass the entire infinite path, their satisfaction values are often derived from a path fragment that precedes a certain state on the path. Therefore, for information synchronization purposes, we must combine the reachability measure of this specific path fragment with the satisfaction value derived from it, in order to obtain accurate and comprehensive measurement information.

References [13,14] introduce the concept of quality constraints, which enhance the expressive power of quantitative temporal logic. Inspired by this work, we introduce quality constraint operators into FLTL, thereby creating FLTL with quality constraints (QFLTL) and accurately integrating path reachability into the path fragment, which is obtained through formula satisfaction values based on preference levels. The syntax and semantics of QFLTL are discussed in Section 2 of the article.

Considering only the quality constraint operators to characterize the varying degrees of importance among sub-property formulas, they are incorporated into the QFLTL formula. When assessing the reachability of the system path (transition process) qualitatively (Boolean), we provide Boolean reachability semantics for QFLTL. We investigate the practicality of QFLTL, analyze its properties, and evaluate its Boolean semantics. These aspects are discussed in Section 3.

We present an automaton construction algorithm for the Boolean reachability semantics of QFLTL and discuss its complexity. Based on this automaton construction method, we investigate searching and decision-making, as well as the logical decision problems pertaining to QFLTL, such as validity, satisfiability (the dual problem to validity), and model checking under Boolean path-reachability semantics. The algorithms for solving these problems are provided, and their complexity is also analyzed. These aspects are discussed in Section 4.

If the property formula is subject to quality constraints while considering the system path (transition process), it becomes quantitatively achievable as well. It is necessary to selectively fuse the satisfaction values of property formulas with path-reachability measures. Based on this premise, it is natural to provide quantitative reachability semantics for QFLTL. However, due to the integration of path-reachability information into the quantitatively reachable QFLTL framework, the calculation of values for its formulas inherently relies on state transition systems. The approach of constructing automata solely based on the QFLTL formula alone is insufficient for solving search and decision problems that involve quantitative path reachability, in contrast to the QFLTL based purely on Boolean reachability. Therefore, the system is confined to scenarios where only infinite loop paths are considered, and the validity, satisfiability, model checking, and other search and decision problems pertaining to the quantitative path-reachability semantics of QFLTL within this specific context are described. Furthermore, the algorithms for solving these problems are presented, and the collective complexity of these algorithms is analyzed. These aspects are discussed in Section 5.

In Section 6 of the article, we present an example of a patient's treatment process. Through this example, we demonstrate that our proposed Boolean reachability semantics and quantitative reachability semantics for QFLTL have a significant practical application value. They can avoid information loss, ensure information synchronization, characterize different weight preferences between path reachability and the satisfaction values of property formulas, and distinguish different weight preferences among property subformulas. At the same time, it has been verified that our proposed algorithms for solving QFLTL search and decision problems are effective.

In Section 7 of the article, a summary of the main contributions of this article is provided, and the potential research directions for future consideration are outlined.

2. The Syntax and Semantics of QFLTL

2.1. Preliminary Knowledge

In Zadeh's fuzzy logic, the classical fuzzy propositional operators encompass "$\neg$" (negation), "$\wedge$" (conjunction), "$\vee$" (disjunction), and "$\longrightarrow$" (implication), among others [15]. To expand this set of classical fuzzy propositional calculus operators and cater to the requirements of QFLTL (quantified fuzzy linear temporal logic), we can introduce a series of propositional quality constraint operators. These newly introduced operators will facilitate the expression of more intricate logical relationships and quality constraints within the framework of QFLTL.

Definition 1 [fuzzy operations]. $\forall x, y \in [0,1]$, *the fuzzy operations 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) $x \bigoplus_{\lambda} y = \lambda \cdot x + (1 - \lambda) \cdot y$.

$\lambda_{cp}(x)$ is called a competence weighting operator, which imposes a weighting constraint on x by multiplying it by λ. $\lambda_{cp}(x)$ maps x to the truth value in the interval $[0, \lambda]$ and even if x attains the maximum truth value of "1", $\lambda_{cp}(x)$ can only attain a value of $\lambda \leq 1$. This effectively reduces the satisfiability value. $\lambda_{ne}(x)$ is called the necessity weighting operator. We consider "true" to be "irrelevant" and "false" to be "critical" (such as in continuous "$\wedge$" operations, where operands can be weighted to avoid information loss caused by logic bridging faults.). $\lambda_{ne}(x)$ maps x to the truth value in the interval $[1 - \lambda, 1]$, and even if x attains the minimum truth value "0", $\lambda_{ne}(x)$ can only obtain $1 - \lambda \geq 0$. This effectively

increases the satisfiability value. The distribution of truth values for propositional variables x_1, x_2, x_3 in $x_1 \wedge x_2 \wedge x_3$ after applying $\lambda_{ne}(x)$ is shown in Table 1. Obviously, the operator $\lambda_{ne}(x)$ increases the true value of the formula, and to some extent, it overcomes the "logic short circuit" caused by continuous " $\wedge$ " operations, refining the original seven "0"s into four "0.2"s, two "0.3"s, and one "0.4"s. The necessity weighting operator enhances the expressive ability of logical formulas in terms of validity.

Table 1. Example of some necessity weighting operators for proposition quality constraints.

x_1	x_2	x_3	$x_1 \wedge x_2 \wedge x_3$	$0.6_{ne}(x_1) \wedge 0.7_{ne}(x_2) \wedge 0.8_{ne}(x_3)$
0	0	0	0	0.2
0	0	1	0	0.3
0	1	0	0	0.2
0	1	1	0	0.4
1	0	0	0	0.2
1	0	1	0	0.3
1	1	0	0	0.2
1	1	1	1	1

The following formalized explanation demonstrates that the λ_{ne} operation fulfills this requirement. "False" is the critical issue, whereas the true value is considered irrelevant. Construct function $f_{[-1,0]} : [0,1] \longrightarrow [-1,0]$. Let

$$f_{[-1,0]}(0) = -1,\ f_{[-1,0]}(1) = 0,\ f_{[-1,0]}(\lambda_{ne}(x)) = \lambda f_{[-1,0]}(x).$$

Then, construct another function $f_{[0,1]} : [-1,0] \longrightarrow [0,1]$. Define a linear transformation as $f_{[0,1]}(x) = 1 + f_{[-1,0]}(x) \Longleftrightarrow f_{[-0,1]}(x) = f_{[0,1]}(x) - 1$;

$$f_{[0,1]}(\lambda_{ne}(x)) = 1 + f_{[-1,0]}(\lambda_{ne}(x)) = 1 + \lambda f_{[-1,0]}(x) = 1 + \lambda\Big(f_{[0,1]}(x) - 1\Big);$$

Then, we get $\lambda_{ne}(x) = 1 + \lambda(x-1) = \lambda x + 1 - \lambda$;

$\lambda_{cf}(x)$ is called the credibility weighting operator. We now expand the range of true values and construct a function. $f_{[-1,1]} : [0,1] \longrightarrow [-1,1]$. Let

$$f_{[-1,1]}(0) = -1, f_{[-1,1]}(1) = 1,\ f_{[-1,1]}\Big(\lambda_{cf}(x)\Big) = \lambda f_{[-1,1]}(x).$$

Then, we construct another function, $f_{[0,1]} : [-1,1] \longrightarrow [0,1]$. Define a linear transformation, as follows.

$$\begin{aligned} f_{[0,1]}(x) &= \Big(1 + f_{[-1,1]}(x)\Big)/2 \Longleftrightarrow f_{[-1,1]}(x) = 2f_{[0,1]}(x) - 1; \\ f_{[0,1]}\Big(\lambda_{cf}(x)\Big) &= \Big(1 + f_{[-1,1]}\Big(\lambda_{cf}(x)\Big)\Big)/2 = \Big(1 + \lambda f_{[-1,1]}(x)\Big)/2 \\ &= \Big(1 + \lambda\Big(2f_{[0,1]}(x) - 1\Big)\Big)/2 = \lambda f_{[0,1]}(x) + (1-\lambda)/2 \end{aligned}$$

Then, we get $\lambda_{cf}(x) = \lambda x + (1-\lambda)/2$;

The operators $\lambda_{cp}(x)$ and $\lambda_{ne}(x)$ are dual. That is, the operator $\lambda_{cp}(x)$ reduces the level of truth. Meanwhile, the operator $\lambda_{ne}(x)$ reduces the level of falsehood. However, the operator $\lambda_{cf}(x)$ reduces the level of falsehood when $x \in [0, 0.5]$ and increases the level of truth when $x \in (0.5, 1]$, resulting in the overall concentration of truth values within interval $[(1-\lambda)/2, (1+\lambda)/2]$. These three operators are linear interpolation operators that formalize quality as a value between zero (false) and one (true). Figure 1 shows a comparison of true value distributions for the fuzzy proposition $x \in [0,1]$ after applying three linear interpolation operators. Here, we set $\lambda = 0.5$.

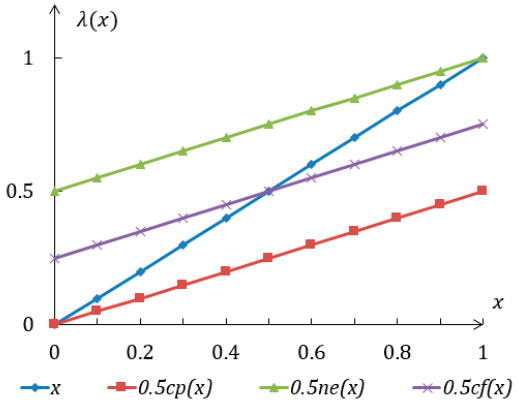

Figure 1. Comparison of true values under the action of linear interpolation operators.

$x \bigoplus_{\lambda} y = \lambda \cdot x + (1-\lambda) \cdot y$ is a weighted average of x and y. λ represents the contribution weight of x to the composite information of x and y, and $1-\lambda$ represents the contribution weight of y to the composite information of x and y. The operator $\bigoplus_{\lambda}$ is used for fusing two pieces of fuzzy information according to their different levels of importance. If $\lambda = 0.5$, then "$\bigoplus_{\lambda}$" is simply denoted as "$\oplus$".

2.2. The Syntax of QFLTL

Definition 2 [structure of QFLTL]. *Assume that AP is a finite set of atomic propositions. $APR \subseteq [0,1] \cap \mathbb{Q}$ is a set of finite fuzzy propositional constants. The QFLTL formula is recursively defined as follows,*

(1) *Atomic proposition $p, p \in AP$;*
(2) *Proposition constant element r, $r \in APR$;*
(3) $\Delta\varphi, \varphi_1 \bigtriangleup_2 \varphi_2, \Delta\varphi, \varphi_1 \bigcup \varphi_2$.

Here,$\mathbb{Q}$ represents the set of rational numbers; φ, φ_1 , and φ_2 are QFLTL formulas. $\bigtriangleup_1 \in \left\{\neg, \lambda_{cp}, \lambda_{ne}, \lambda_{cf}\right\}$ is a unary fuzzy propositional logic operator, and $\bigtriangleup_2 \in \{\wedge, \vee, \longrightarrow, \bigoplus_{\lambda}\}$ is a binary fuzzy propositional logic operator. $\Delta \in \{\bigcirc, \Diamond, \Box\}$ is a unary temporal logic operator. The formulas obtained by finite recursive nesting of the above three rules are all QFLTL formulas.

2.3. Semantics and Practical Examples of QFLTL

The semantics of QFLTL is a fuzzy function, given a sequence of fuzzy propositions π and a QFLTL formula φ, and the semantics map π and φ to the fuzzy satisfaction value of φ on π.

The fuzzy proposition sequence π can characterize a path in a physical symbol system. Below is a quantitative definition of a physical symbol system—fuzzy Kripke structures (FKSs) [7,8].

Definition 3 [Fuzzy Kripke Structures (FKSs)]. *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 4 [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.

Definition 5 [Boolean reachable semantics of QFLTL]. *The Boolean reachable semantics* $\|\cdot\|$ *maps an infinite path* $\pi \in \left([0,1]^{AP}\right)^\omega$ *and a QFLTL formula* φ *to the fuzzy satisfaction value* $\|\pi, \varphi\|$ *of* φ *on* π, *which is defined as follows.*

(1) $\left\|\pi^i, r\right\| = r$;
(2) $\left\|\pi^i, p\right\| = L(\pi_i)(p)$;
(3) $\left\|\pi^i, \bigtriangleup_1(\varphi)\right\| = \bigtriangleup_1(\left\|\pi^i, \varphi\right\|)$;
(4) $\left\|\pi^i, \varphi_1 \bigtriangleup_2 \varphi_2\right\| = \left\|\pi^i, \varphi_1\right\| \bigtriangleup_2 \left\|\pi^i, \varphi_2\right\|$;
(5) $\left\|\pi^i, \bigcirc\varphi\right\| = \left\|\pi^{i+1}, \varphi\right\|$;
(6) $\left\|\pi^i, \diamond\varphi\right\| = \bigvee\limits_{j \geq i} \left\|\pi^j, \varphi\right\|$;
(7) $\left\|\pi^i, \Box\varphi\right\| = \bigwedge\limits_{j \geq i} \left\|\pi^j, \varphi\right\|$;
(8) $\left\|\pi^i, \varphi_1 \bigcup \varphi_2\right\| = \bigvee\limits_{j \geq i} (\left\|\pi^j, \varphi_2\right\| \wedge \bigwedge\limits_{i \leq k < j} \left\|\pi^k, \varphi_1\right\|)$.

Definition 6 [quantitatively reachable semantics of QFLTL]. *Suppose* $M = (S, I, \delta, AP, L)$ *is an FKS, then* $\pi = \pi_0, \pi_1, \cdots, \pi_i, \pi_{i+1}, \cdots \in S^\omega$ *is any path in* M; $\circ \in \{\bigwedge, \cdot, \bigoplus_\lambda\}$ *is a fuzzy synthesis operation. The quantitative reachability semantics* $[\![\cdot]\!]$ *maps an infinite path* $\pi \in \left([0,1]^{AP}\right)^\omega$ *and a QFLTL formula* φ *to the fuzzy satisfaction value* $[\![\pi, \varphi]\!]$ *of* φ *on* π, *which is defined as follows.*

(1) $[\![\pi^i, r]\!] = \delta_i^*(\pi) \circ r$;
(2) $[\![\pi^i, p]\!] = \delta_i^*(\pi) \circ L(\pi_i)(p)$;
(3) $[\![\pi^i, \bigtriangleup_1(\varphi)]\!] = \bigtriangleup_1([\![\pi^i, \varphi]\!])$;
(4) $[\![\pi^i, \varphi_1 \bigtriangleup_2 \varphi_2]\!] = [\![\pi^i, \varphi_1]\!] \bigtriangleup_2 [\![\pi^i, \varphi_2]\!]$;
(5) $[\![\pi^i, \bigcirc\varphi]\!] = [\![\pi^{i+1}, \varphi]\!]$;
(6) $[\![\pi^i, \diamond\varphi]\!] = \bigvee\limits_{j \geq i} [\![\pi^j, \varphi]\!]$;
(7) $[\![\pi^i, \Box\varphi]\!] = \bigwedge\limits_{j \geq i} [\![\pi^j, \varphi]\!]$;
(8) $[\![\pi^i, \varphi_1 \bigcup \varphi_2]\!] = \bigvee\limits_{j \geq i} \left([\![\pi^j, \varphi_2]\!] \wedge \bigwedge\limits_{i \leq k < j} [\![\pi^k, \varphi_1]\!] \right)$.

Note that the quantitative semantic $[\![\cdot]\!]$ is recursively defined. Formally, it appears that, except when φ is a fuzzy constant $r \in APR$ or an atomic proposition $p \in AP$, $[\![\cdot]\!]$ synthesizes path-reachability information. However, no explicit path-reachability information is directly extracted in other situations. Nevertheless, when the recursion terminates with φ being a fuzzy constant r or an atomic proposition p, in essence, regardless of the intermediate forms taken, the information is synthesized through recursive iterations. If each form were to be considered separately, it would in fact lead to redundancy in the representation of reachability information.

The principle of multiplication is embodied by "$\cdot$". To obtain the quantitative reachability semantic measure $[\![\pi^i, \varphi]\!]$ of the QFLTL formula φ on path $\pi^i = \pi_i, \pi_{i+1}, \cdots \in S^\omega$, we first compute the reachability $\delta_i^*(\pi)$ of the prefix path $\pi_0, \pi_1, \cdots, \pi_i$ and, then, multiply it by the satisfaction value $\|\pi^i, \varphi\|$ of the formula. "$\bigoplus_\lambda$" embodies the idea of weighted average, considering the fusion of path reachability and formula satisfaction values according to different weights.

If the propositional quality constraint operators $\lambda_{cp}, \lambda_{ne}, \lambda_{cf}$, and $\bigoplus_\lambda$ are removed, and only the "$\wedge$" operation is utilized during information synthesis, then QFLTL simplifies into FLTL, as described in references [10,11]. Clearly, FLTL is a subset of QFLTL.

Now, we will present some examples of the system properties described using QFLTL to illustrate the practicality of QFLTL.

Example 1. *Figure 2 shows a 4-state FKS.*

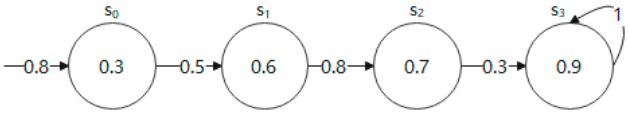

Figure 2. A 4-state FKS.

$AP = \{a\}, \mathrm{S} = \{s_0, s_1, s_2, s_3\}, I = (0.8, 0, 0, 0)^{\mathrm{T}}, I(s_0) = 0.8, L(s_0)(a) = 0.3, \cdots, L(s_3)(a) = 0.9, \delta(s_0, s_1) = 0.5, \cdots, \delta(s_3, s_3) = 0.8$. There is only one infinite path; that is $\pi = s_0, s_1, s_2, {s_3}^\omega, \delta_0^*(\pi) = 0.8, \delta_1^*(\pi) = 0.5, \delta_2^*(\pi) = 0.5$, when $i \geq 3, \delta_3^*(\pi) = 0.3$,

If "." is taken as the information fusion operator, then,

$$[\![\pi^i, \Diamond a]\!] = \bigvee_{i \geq 0} [\![\pi^i, a]\!] = \bigvee_{i \geq 0} \delta_i^*(\pi) \cdot \pi_i(a)$$
$$= (0.3 \times 0.8) \vee (0.6 \times 0.5) \vee (0.7 \times 0.5) \vee (0.9 \times 0.3) \vee (0.9 \times 0.3) \vee \cdots = 0.35.$$

If it is believed that the formula satisfies an importance that is three times greater than that of the value formula path reachability, we take $\bigoplus_{0.25}$ as the information fusion operator. Then,

$$[\![\pi, (\Diamond a)]\!] = \bigvee_{i \geq 0} [\![\pi^i, a]\!] = \bigvee_{i \geq 0} \delta_i^*(\pi) \bigoplus_{0.25} \pi_i(a)$$
$$= (0.8 \times 0.25 + 0.3 \times 0.75) \vee (0.5 \times 0.25 + 0.6 \times 0.75) \vee (0.5 \times 0.25 + 0.7 \times 0.75)$$
$$\vee (0.3 \times 0.25 + 0.9 \times 0.75) \vee (0.3 \times 0.25 + 0.9 \times 0.75) \vee \cdots = 0.75$$

At this point, the value of $[\![\pi, \Diamond a]\!]$ is obtained along the path fragment cluster $\{\pi = s_0, s_1, s_2, s_3^i \mid i \geq 1\}$.

Example 2. *Consider a scheduler that accepts requests and generates authorizations. Use the QFLTL formula*

$$\varphi = \Box\Big(request \longrightarrow \Big(grant \bigoplus\nolimits_{3/4} \bigcirc grant\Big)\Big) \wedge \neg(4/5)_{cp}(\neg request)$$

to characterize the following temporal property.

During the runtime of the scheduler, it always ensures that, once a process requests a critical resource (*request*), it will be granted in the future or in the immediate next state (*grant*). If the process does not issue any authorization requests throughout the entire scheduling process, it is considered to fulfill a condition related to a value of 3/4, but the exact formulation of this condition in temporal logic would depend on the context and specific requirements.

Assuming $\pi \in \left([0,1]^{AP}\right)^{\omega}$ is an arbitrary path, it is used to characterize the process scheduling process. Below, we will analyze the satisfaction values of the formula on different paths for π.

(1) If the following LTL formula is used, it is difficult to characterize the above temporal properties accurately;

$$\Box(request \rightarrow \Diamond(grant \vee \bigcirc grant) \wedge (\Box \neg request))$$

(2) Obviously, this LTL formula struggles to distinguish between situations where no request has been made and situations where requests are authorized at a future system time or at a subsequent system time;

(3) To characterize this temporal property using the aforementioned QFLTL formula, we can express it as follows. When no request is issued at all, the satisfaction value of the formula is one-fifth. After a request is issued but authorization is never granted, the satisfaction value is zero. After a request is issued, if there exists a future adjacent state pair where both states grant authorization, the satisfaction value is one. If in a state pair, the first state grants authorization but the second does not, the satisfaction value is three-fourths. And if in a state pair, the first state does not grant authorization but the second does, the satisfaction value is one-fourth.

Obviously, the five satisfaction values correspond to the five possible scenarios that may arise during the process scheduling procedure. Therefore, QFLTL has a stronger expressive ability than LTL.

3. The Properties and Boolean Transformations of QFLTL

3.1. Relationship between QFLTL Formulas

Definition 7 [the relations between QFLTL formulas]. *Let φ_1 and φ_2 be the QFLTL formulas. $M = (S, I, \delta, AP, L)$ is an FKS. $Path(M) = \{\pi | \pi \in S^{\omega}, I(\pi_0) > 0\}$ is the set of infinite paths in M, and $\sim \in (>, <, \geq, \leq, \neq, =)$ is a relational operator.*

(1) If $\forall \pi \in \left([0,1]^{AP}\right)^{\omega}$, $\|\pi, \varphi_1\| \sim \|\pi, \varphi_2\|$ then we say that φ_1 and φ_2 satisfy the relationship "$\sim$", which is denoted as $\varphi_1 \sim \varphi_2$;

(2) If $\forall \pi \in Path(M)$, $[\![\pi, \varphi_1]\!] \sim [\![\pi, \varphi_2]\!]$ then we say that φ_1 and φ_2 satisfy the relationship "$\sim$" in M, which is denoted as $\varphi_1 \sim_M \varphi_2$.

Proposition 1 [equivalent calculus of QFLTL]. *If φ_1 and φ_2 are QFLTL formulas, and $M = (S, I, \delta, AP, L)$ is an FKS, then,*

(1) $\varphi = \neg\neg\varphi$;
(2) $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2)$; $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$;
(3) $\varphi_1 \longrightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$;
(4) $\Diamond\varphi = True \cup \varphi$;
(5) $\Box\varphi = \neg(\Diamond\neg\varphi)$;
(6) $\Diamond\varphi = \varphi \vee \bigcirc\Diamond\varphi$
(7) $\Box\varphi = \varphi \wedge \bigcirc\Box\varphi$
(8) $\varphi_1 \bigcup \varphi_2 = \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \bigcup \varphi_2)$;
(9) $\varphi =_M \neg\neg\varphi$;
(10) $\varphi_1 \wedge \varphi_2 =_M \neg(\neg\varphi_1 \vee \neg\varphi_2)$; $\varphi_1 \vee \varphi_2 =_M \neg(\neg\varphi_1 \wedge \neg\varphi_2)$;
(11) $\varphi_1 \longrightarrow \varphi_2 \geq_M \neg\varphi_1 \vee \varphi_2$;
(12) $\Diamond\varphi =_M True \cup \varphi$;
(13) $\Box\varphi =_M \neg(\Diamond\neg\varphi)$;
(14) $\Diamond\varphi =_M \varphi \vee \bigcirc\Diamond\varphi$;
(15) $\Box\varphi =_M \varphi \wedge \bigcirc\Box\varphi$;
(16) $\varphi_1 \bigcup \varphi_2 = \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \bigcup \varphi_2)$.

Proof. According to Definition 5, it is straightforward to prove the conclusion of Proposition 1.□

Proposition 2. *The set* $\left\{\neg, \lambda_{cp}, \lambda_{ne}, \lambda_{cf}, \vee, \bigoplus_{\lambda}, \bigcirc, \diamond, \bigcup\right\}$ *constitutes a functionally complete set of QFLTL operators.*

According to Definitions 5 and 6, the QFLTL semantics assign a fuzzy satisfaction value to a given path π and a QFLTL formula φ. Thus, the value of the QFLTL formula depends on both φ itself and the specified path π. Subsequently, through meticulous analysis and the summarization of φ and π, the properties of QFLTL and the Boolean method are elucidated. This serves as a prerequisite for the subsequent automation and reasoning processes.

3.2. Boundedness of the Range of QFLTL Formulas

Definition 8 [spanning tree of QFLTL formulas]. *The spanning tree of a QFLTL formula φ is recursively defined as follows.*

(1) A proposition constant r or an atomic proposition p in φ is characterized as a leaf node of the spanning tree of φ;

(2) Each operator of φ corresponds to a subtree of the spanning tree of φ, where the operator serves as the root node of its subtree, and the subformulas associated with this operator are the branches of the subtree.

Definition 9 [length of QFLTL formulas]. *The number of nodes in the spanning tree of a QFLTL formula φ is called the length of φ, which is denoted as $|\varphi|$.*

Definition 10 [calculated base and cardinality]. *Given a path of* $\pi = \pi_0, \pi_1, \cdots, \pi_i, \pi_{i+1}, \cdots \in \left([0,1]^{AP}\right)^{\omega}$, *the set,* $B_\pi = \left\{\pi_i | i \in \mathbb{N}, \pi_i \in \pi, \forall\, i \neq j,\ \pi_i \neq \pi_j.\right\}$ *is called the base of* π, *and* $|\pi| = |B_\pi| \in \mathbb{N}^+$ *is called the cardinality of* π.

A path with limited cardinality is called a measurable path. If Π is a set of measurable paths, then $\bigvee\limits_{\pi \in \Pi} |B_\pi|$ is called the cardinality of Π.

Obviously, the basis of a finite path is finite. In fact, the basis of an infinite path can also be finite. The following are two important definitions related to infinite paths with finite bases.

Definition 11 [lasso path] *If a path* $\pi \in \left([0,1]^{AP}\right)^{\omega}$ *has the form*

$$\pi = \pi_0, \pi_1, \cdots, \pi_{\mu-1}, (\pi_{\mu}, \pi_{\mu+1}, \cdots \pi_{\mu+v-1})^{\omega}$$

where $\mu \in \mathbb{N}, v \in \mathbb{N}^+$, *then* π *is called a lasso path.*

Definition 12 [pure lasso path]. *Let* $\pi = \pi_0, \pi_1, \cdots, \pi_{\mu-1}, (\pi_{\mu}, \pi_{\mu+1}, \cdots \pi_{\mu+v-1})^{\omega}$ *be a lasso path, where* $\mu \in \mathbb{N}$, *and* $v \in \mathbb{N}^+$ *If* $\pi_0, \pi_1, \cdots, \pi_{\mu-1}, \pi_{\mu}, \pi_{\mu+1}, \cdots \pi_{\mu+v-1}$ *are distinct from each other, then* π *is called a pure lasso path.*

Proposition 3 [base of lasso path]. *The base of the lasso path*

$$\pi = \pi_0, \pi_1, \cdots, \pi_{\mu-1}, (\pi_{\mu}, \pi_{\mu+1}, \cdots \pi_{\mu+v-1})^{\omega}$$

is $|\pi| \leq \mu + v$, *if and only if* π *is a pure lasso path.*

Theorem 1 [bounded range of QFLTL formula]. *Let φ be a QFLTL formula and $M = (S, I, \delta, AP, L)$ be an FKS. $V[\varphi] = \left\{ \|\pi, \varphi\| \middle| \pi \in \left([0,1]^{AP}\right)^{\omega} \right\}$ be the Boolean reachable semantic range, $V[\![\varphi]\!] = \{[\![\pi, \varphi]\!] | \pi \in S^{\omega}\}$ be the quantitative reachable semantic range, and m be the cardinality of the measurable path set in $\left([0,1]^{AP}\right)^{\omega}$. Then, $\left|V[\varphi]\right| \leq m^{|\varphi|}$ and $|V[\![\varphi]\!]| \leq |S|^{|\varphi|+2}$.*

Proof. First, we prove that $\left|\mathrm{V}[\varphi]\right| \leq \mathrm{m}^{|\varphi|}$. This proof is induced based on the structure of φ.

(1) If $\varphi = r, p, |\mathrm{V}[\varphi]| = 1 \leq m = m^{|\varphi|}$;
(2) If $\varphi = \bigtriangleup_1 \phi, \bigcirc \phi, \Diamond \phi, \Box \phi, |\mathrm{V}[\varphi]| = |\mathrm{V}[\phi]|$, based on the inductive hypotheses, we obtain $|\mathrm{V}[\varphi]| = |\mathrm{V}[\phi]| \leq m^{|\phi|} \leq m^{|\phi|+1} = m^{|\varphi|}$;
(3) If $\varphi = \varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2, \varphi_1 \longrightarrow \varphi_2, \varphi_1 \bigoplus_{\lambda} \varphi_2, \varphi_1 \bigcup \varphi_2 \mathrm{V}[\varphi] \subseteq \mathrm{V}[\varphi_1] \cup \mathrm{V}[\varphi_2], |\mathrm{V}[\varphi]| \leq |\mathrm{V}[\varphi_1]| + |\mathrm{V}[\varphi_2]|, |\mathrm{V}[\varphi]| \leq m^{|\varphi_1|} + m^{|\varphi_2|} \leq m^{|\varphi_1|+|\varphi_2|} \leq m^{|\varphi_1|+|\varphi_2|+1} = m^{|\varphi|}$.
Next, we prove that $|\mathrm{V}[\![\varphi]\!]| \leq |S|^{|\varphi|+2}$.
From the definition of δ_i^*, it is easy to deduce that $\left|\{\delta_i^* | i \in \mathbb{N}\}\right| \leq |S \times S| = |S|^2$. Based on the structure of φ, the following is the proof.
(4) When "$\circ$" takes "$\wedge$", $\mathrm{V}[\![\varphi]\!] \subseteq \mathrm{V}[\varphi] \cup \{\delta_i^* | i \in \mathbb{N}\}, \left|\mathrm{V}[\![\varphi]\!]\right| \leq |\mathrm{V}[\varphi]| + |S|^2 \leq |S|^{|\varphi|} \cdot |S|^2 \leq |S|^{|\varphi|+2}$;
(5) When "$\circ$" takes "$\cdot$" or "$\bigoplus_{\lambda}$", $\mathrm{V}[\![\varphi]\!] \subseteq \{\delta_i^* | i \in \mathbb{N}\} \times \mathrm{V}[\varphi], \left|\mathrm{V}[\![\varphi]\!]\right| \leq |\mathrm{V}[\varphi]| \cdot |S|^2 = |S|^{|\varphi|} \cdot |S|^2 \leq |S|^{|\varphi|+2}$. □

3.3. Boolean Characterization of QFLTL Formula

Next, using the fuzzy predicate $P \subseteq [0,1] \cap \mathbb{Q}$ ($\mathbb{Q}$ is a set of rational numbers.) to qualitatively partition the satisfaction values of the QFLTL formula φ. The aim is to characterize the qualitative satisfaction property of φ.

Lemma 1. *If P is a fuzzy predicate and $P = [\alpha, \beta]$ (where P can be an open interval or a semi-open or semi-closed interval), and $\lambda \in (0,1)$ is a fuzzy number, then we have,*

(1) $\left\|\pi, \lambda_{cp}(\varphi)\right\| \in [\alpha, \beta]$, *if and only if* $\|\pi, \varphi\| \in P_{cp} = [\alpha/\lambda, \beta/\lambda]$;
(2) $\|\pi, \lambda_{ne}(\varphi)\| \in [\alpha, \beta]$ *if and only if* $\|\pi, \varphi\| \in P_{ne} = [(\alpha + \lambda - 1)/\lambda, (\beta + \lambda - 1)/\lambda]$;
(3) $\left\|\pi, \lambda_{cf}(\varphi)\right\| \in [\alpha, \beta]$ *if and only if* $\|\pi, \varphi\| \in P_{cf} = [(2\alpha + \lambda - 1)/2\lambda, (2\beta + \lambda - 1)/2\lambda]$;
(4) $\|\pi, \varphi_1 \bigoplus_{\lambda} \varphi_2\| \in [\alpha, \beta]$, *if and only if,* $\exists d_1, d_2,\ d_1 = \|\pi, \varphi_1\| \in V[\varphi_1]$, $d_2 = \|\pi, \varphi_2\| \in V[\varphi_2]$

$d_1 \bigoplus_{\lambda} d_2 \in [\alpha, \beta]$.

Proof. According to the definition of $\|\cdot\|$ given in Definition 5, the result is obvious. □

Theorem 2 [Boolean transformation of QFLTL]. *Let φ be a QFLTL formula, π be an infinite path, and P be a predicate over the satisfaction values of QFLTL formulas. There exists an LTL formula $Bool(\varphi, P)$ such that $\|\pi, \varphi\| \in P$ if and only if $\pi \vDash Bool(\varphi, P)$.*

Proof. The construction of $B\mathrm{ool}(\varphi, P)$ is as follows.

(1) If $\varphi = r \in [0,1], \mathrm{Bool}(\varphi, P) = \begin{cases} True & r \in P \\ False & r \notin P \end{cases}$;
(2) If $\varphi = p \in AP, \mathrm{Bool}(p, P) = \begin{cases} True & [0,1] \subseteq P \\ p & \pi_0(p) \in P \\ \neg p & \pi_0(p) \notin P \\ False & [0,1] \cap P = \emptyset \end{cases}$;
(3) If $\varphi = \neg\phi, \mathrm{Bool}(\varphi, P) = \neg \mathrm{Bool}(\phi, P)$;
(4) If $\varphi = \lambda_{cp}(\phi), \mathrm{Bool}\left(\lambda_{cp}(\phi), P\right) = \mathrm{Bool}\left(\phi, P_{cp}\right)$;

(5) If $\varphi = \lambda_{ne}(\phi)$, $Bool(\lambda_{ne}(\phi), P)$=$Bool(\phi, P_{ne})$;
(6) If $\varphi = \lambda_{cf}(\phi)$, $Bool\left(\lambda_{cf}(\phi), P\right)$=$Bool\left(\phi, P_{cf}\right)$;
(7) If $\varphi = \varphi_1 \bigoplus_\lambda \varphi_2$, $Bool(\varphi, P) = \bigvee_{d_1 \in V[\varphi_1], d_2 \in V[\varphi_2], d_1 \bigoplus_\lambda d_2 \in [\alpha,\beta]} Bool(\varphi_1, d_1) \wedge Bool(\varphi_2, d_2)$;
(8) If $\varphi = \varphi_1 \vee \varphi_2$, $Bool(\varphi_1 \vee \varphi_2, P) = Bool(\varphi_1, P) \vee Bool(\varphi_2, P)$;
(9) If $\varphi = \bigcirc\phi$, $Bool(\bigcirc\phi, \mathrm{P}) = \bigcirc Bool(\phi, P)$;
(10) If $\varphi = \Diamond\phi$, $Bool(\Diamond\phi, \mathrm{P}) = \Diamond Bool(\phi, P)$;
(11) If $\varphi = \Box\phi$, $Bool(\Box\phi, P) = \Box Bool(\phi, P)$;
(12) If $\varphi = \varphi_1 \bigcup \varphi_2$ $Bool(\varphi_1 \bigcup \varphi_2, P) = \bigvee_{c \in P \bigcap V[\varphi]} ((Bool(\varphi_1, [c,1]) \bigcup Bool(\varphi_2, [c,1]))$

$$\wedge\neg(Bool(\varphi_1, [0,c)) \bigcup Bool(\varphi_2, [0,c)))).$$

According to the above construction of $Bool(\phi, P)$, it is straightforward to prove cases (1) through (11), and here, we will prove a relatively complex case involving $\varphi = \varphi_1 \bigcup \varphi_2$.

$$\begin{aligned}
&\|\pi, \varphi\| = c \in P \cap V[\varphi] \Longleftrightarrow \bigvee_{i\geq 0}\left(\|\pi^i, \varphi_2\| \wedge \bigwedge_{0\leq j<i}\|\pi^j, \varphi_1\|\right) = c \\
&\Longleftrightarrow \left(\bigvee_{i\geq 0}\left(\|\pi^i, \varphi_2\| \wedge \bigwedge_{0\leq j<i}\|\pi^j, \varphi_1\|\right) \geq c\right) \wedge \neg\left(\bigvee_{i\geq 0}\left(\|\pi^i, \varphi_2\| \wedge \bigwedge_{0\leq j<i}\|\pi^j, \varphi_1\|\right) < c\right) \\
&\Longleftrightarrow \left(\bigvee_{i\geq 0}\left(\|\pi^i, \varphi_2\| \wedge \bigwedge_{0\leq j<i}\|\pi^j, \varphi_1\|\right) \in [c,1)\right) \wedge \neg\left(\bigvee_{i\geq 0}\left(\|\pi^i, \varphi_2\| \wedge \bigwedge_{0\leq j<i}\|\pi^j, \varphi_1\|\right) \in [0,c)\right) \\
&\Longleftrightarrow (\bigvee_{i\geq 0}(\pi^i \vDash Bool(\varphi_1, [c,1]) \wedge \bigwedge_{0\leq j<i} \pi^j \vDash Bool(\varphi_2, [c,1])) \\
&\wedge\neg(\bigvee_{i\geq 0}(\pi^i \vDash Bool(\varphi_1, [0,c)]) \wedge \bigwedge_{0\leq j<i} \pi^j \vDash Bool(\varphi_2, [0,c)))) \\
&\Longleftrightarrow \pi \vDash (Bool(\varphi_1, [c,1]) \bigcup Bool(\varphi_2, [c,1])) \wedge \neg(Bool(\varphi_1, [0,c)) \bigcup Bool(\varphi_2, [0,c))). \square
\end{aligned}$$

According to Theorem 2, the quantitative reasoning problem of QFLTL formulas can be reduced to the classical LTL reasoning framework for solving.

4. Search and Decision Problems for Boolean Reachable Semantics in QFLTL

Validity, satisfiability, model checking, and other core issues in search and decision-making are addressed within the QFLTL framework, along with the solutions to these issues.

4.1. The Automaton Representation of Boolean Reachable Semantics in QFLTL

Within the inferential theoretical frameworks of classic LTL, PLTL, and PoLTL, there exist theories and applications that construct an automaton $A_{\varphi,P}$ capable of qualitatively satisfying the predicate P with the formula φ. Subsequently, we solve related search and decision problems by assessing the emptiness of $A_{\varphi,P}$ or the product of the physical symbol system Γ, which is obtained from the current application modeling and $A_{\varphi,P}$. Next, we will extend this theory to the QFLTL framework.

Theorem 3 [constructing QFLTL formula as an automaton]. *Let φbe a QFLTL formula, $\pi \in \left([0,1]^{AP}\right)^\omega$ be a measurable infinite path, and $P \subseteq [0,1] \cap \mathbb{Q}$ be a fuzzy predicate. Let m be the cardinality of the set of all measurable paths. There exists a NGBA $A_{\varphi,P}$, such that $\|\pi, \varphi\| \in P$ if and only if $\pi \in L(A_{\varphi,P})$. The number of states of $A_{\varphi,P}$ does not exceed $m^{|\varphi|^2}$. The index (i.e., the number of Büchi acceptance sets) of $A_{\varphi,P}$ does not exceed $|\varphi|$.*

Proof. The set of sub formulas of φ is denoted as $cl(\varphi)$, and

$$C_\varphi = \{g | g : cl(\varphi) \to [0,1], \forall\phi \in cl(\varphi), g(\varphi) \in \mathrm{V}[\varphi]\}.$$

Construct an NGBA $A_{\varphi,P} = \{2^{AP}, Q, \delta, Q_0, F\}$, where 2^{AP} is the input alphabet (set of symbols). The state space of $A_{\varphi,P}$ is $Q \in C_\varphi$. The transition function $\delta : Q \times 2^{AP} \to Q$ determines the next state, given the current state and input symbol. If the current state is $g_i \in Q$, it transitions to the next state g_{i+1} upon inputting $\sigma_i \in 2^{AP}$. The initial state set of $A_{\varphi,P}$ is $Q_0 = \{g_0\}$. Where g_0 is a specific function in C_φ.F is the set of final states, where each $\varphi_1 \bigcup \varphi_2 \in cl(\varphi)$ contributes a subset of final states

$$F_{\varphi_1 \cup \varphi_2} = \left\{ g \middle| g(\varphi_2) = g\left(\varphi_1 \bigcup \varphi_2\right) \right\} \text{ to } F.$$

For all $\phi \in cl(\varphi)$, $A_{\varphi,P}$ satisfies the following consistency constraints.

(1) Consistency constraint for propositional logic (CP rules);

– $CP - r$ The consistency of fuzzy constant propositions.

when $\phi = r \in [0,1], g_i(r) = \begin{cases} r & r \in P \\ 0 & r \notin P \end{cases}$;

– $CP - AP$ The consistency of fuzzy atomic propositions.

When $\phi = p \in AP, g_i(p) = \begin{cases} \pi_i(p) & \pi_i(p) \in P \\ 0 & \pi_i(p) \notin P \end{cases}$;

– $CP - \neg$ When $\phi = \neg\psi, g_i(\phi) = \neg g_i(\psi) = 1 - g_i(\psi)$;
– $CP - \lambda_{cp}$ When $\phi = \lambda_{cp}(\psi), g_i(\phi) = \lambda \cdot g_i(\psi)$;
– $CP - \lambda_{ne}$ When $\phi = \lambda_{ne}(\psi), g_i(\phi) = \lambda \cdot g_i(\psi) + 1 - \lambda$;
– $CP - \lambda_{cf}$ When $\phi = \lambda_{cf}(\psi), g_i(\phi) = \lambda \cdot g_i(\psi) + (1 - \lambda)/2$;
– $CP - \wedge$ When $\phi = \psi_1 \wedge \psi_2, g_i(\phi) = g_i(\psi_1) \wedge g_i(\psi_2)$;
– $CP - \vee$ When $\phi = \psi_1 \vee \psi_2, g_i(\phi) = g_i(\psi_1) \vee g_i(\psi_2)$;
– $CP - \circ$ When $\phi = \psi_1 \circ \psi_2, g_i(\phi) = g_i(\psi_1) \circ g_i(\psi_2)$.

(2) Temporal Consistency Constraint. (CT Rules).

– —$CP - \sigma.\Sigma = (p | p \in AP, g_i(p) > 0)$;
– —$CP - \bigcirc$ When $\phi = \bigcirc\psi$ When $g_i(\phi) = g_{i+1}(\psi)$;
– —$CP - \Diamond$ When $\phi = \Diamond\psi$ $g_i(\phi) = g_i(\psi) \vee g_{i+1}(\psi)$;
– —$CP - \Box$ When $\phi = \Box\psi$ $g_i(\phi) = g_i(\psi) \wedge g_{i+1}(\psi)$;
– —$CP - \bigcup$ When $\phi = \psi_1 \bigcup \psi_2, g_i(\phi) = g_i(\psi_2) \vee (g_i(\psi_1) \wedge (g_{i+1}(\psi_1) \bigcup g_{i+1}(\psi_2)))$.

Next, we prove that "$\|\pi, \varphi\| \in P \Longrightarrow \pi \in L(A_{\varphi,P})$".

For all $\phi \in cl(\varphi)$, let $g_i(\phi) = \|\pi^i, \phi\|$. According to Definition 5, it can be proven that the running $\rho = g_0, g_1, \ldots$ satisfies all the consistency mentioned above. Here is a relatively complex "$CP - \bigcup$", as demonstrated by the following example.

$$\begin{aligned} g_i(\varphi_1 \bigcup \varphi_2) = \|\pi^i, \varphi_1 \bigcup \varphi_2\| &= \bigvee_{j \geq i} \left(\|\pi^j, \varphi_2\| \wedge \bigwedge_{i \leq k < j} \left\|\pi^k, \varphi_1\right\| \right) \\ &= g_i(\varphi_2) \vee (g_i(\varphi_1) \wedge g_{i+1}(\varphi_1 \bigcup \varphi_2)). \end{aligned}$$

Next, it is necessary to prove that there exists an infinite number of states in $F_{\varphi_1 \cup \varphi_2} = \{g | g(\varphi_2) = g(\varphi_1 \bigcup \varphi_2)\}$, suchthat $g \in \{g_0, g_1, \ldots\}$. To do this, let us first prove that $\forall i \in \mathbb{N}$,

$$\left\|\pi^i, \varphi_2\right\| \leq \left\|\pi^i, \varphi_1 \bigcup \varphi_2\right\| \tag{1}$$

$$\left\|\pi^i, \varphi_1 \bigcup \varphi_2\right\| = \left\|\pi^i, \varphi_2\right\| \vee \left(\left\|\pi^i, \varphi_1\right\| \wedge \bigvee_{j \geq i+1} \left(\left\|\pi^j, \varphi_2\right\| \wedge \bigwedge_{i \leq k < j} \left\|\pi^k, \varphi_1\right\| \right) \right) \geq \left\|\pi^i, \varphi_2\right\|.$$

Furthermore, we prove that $\exists j \in \mathbb{N}$, such that $\left\|\pi^j, \varphi_2\right\| \geq \left\|\pi^j, \varphi_1 \bigcup \varphi_2\right\|$. According to Theorem 4, since $V[\varphi_2]$ is finite, $\exists j \in \mathbb{N}$, such that $\left\|\pi^j, \varphi_2\right\| = \bigvee\limits_{k \geq j} \left\|\pi^k, \varphi_2\right\|$. This leads to the following conclusion.

$$\left\|\pi^j, \varphi_2\right\| = \bigvee_{k \geq j} \left\|\pi^k, \varphi_2\right\| \geq \bigvee_{k \geq j} \left(\left\|\pi^j, \varphi_2\right\| \wedge \bigwedge_{j \leq t < k} \left\|\pi^k, \varphi_1\right\| \right) = \left\|\pi^j, \varphi_1 \bigcup \varphi_2\right\|.$$

Since the premise of Equation (1) is $\forall j \in \mathbb{N}$, it can be obtained that,

$$\left\|\pi^j, \varphi_2\right\| \leq \left\|\pi^j, \varphi_1 \bigcup \varphi_2\right\|. \tag{2}$$

By integrating (1) and (2), it can be concluded that

$$\forall i \in \mathbb{N}, \exists j \geq i \left\|\pi^j, \varphi_2\right\| = \left\|\pi^j, \varphi_1 \bigcup \varphi_2\right\|.$$

Furthermore, it can be concluded that $g_j(\varphi_2) = g_j(\varphi_1 \bigcup \varphi_2)$. By this approach, we have $\forall i \in \mathbb{N}, \exists j \geq i$, $g_j \in F_{\varphi_1 \bigcup \varphi_2} = \{g | g(\varphi_2) = g(\varphi_1 \bigcup \varphi_2)\} \subseteq F$.

In summary, $\rho = g_0, g_1, \ldots$ is an acceptable run in $A_{\varphi,P}$.

On the other hand, it needs to prove that $\|\pi, \varphi\| \in P \Longleftarrow \pi \in L(A_{\varphi,P})$.

Let $= g_0, g_1, \ldots$ be an acceptable run on $\pi \in L(A_{\varphi,P})$ in $A_{\varphi,P}$.$\forall i \in \mathbb{N}$,$\forall \phi \in cl(\varphi)$. We now prove that $\left\|\pi^i, \varphi\right\| = g_i(\varphi)$. This conclusion is easily proven based on the structure of φ and the construction of $A_{\varphi,P}$. For a more complex situation, that is, when $\phi = \varphi_1 \bigcup \varphi_2$, we prove that,

$$\left\|\pi^i, \phi\right\| = g_i\left(\varphi_1 \bigcup \varphi_2\right). \square \tag{3}$$

To prove formula (3), we first need to prove the following two lemmas.

Lemma 2. *If* $\phi = \varphi_1 \bigcup \varphi_2 \in cl(\varphi)$, $g_l \in F_\varphi$, *then*

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) = \bigvee_{j \geq l} \left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1) \right) = \left\|\pi^l, \varphi_1 \bigcup \varphi_2\right\|.$$

Proof. First, we prove that

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) \leq \bigvee_{j \geq l} \left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1) \right). \tag{4}$$

$$\bigvee_{j \geq l} \left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1) \right) = g_l(\varphi_2) \vee \bigvee_{j \geq l+1} \left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1) \right) \geq g_l(\varphi_2)$$

Since $g_l \in F_\varphi$, then $\bigvee\limits_{j \geq l} \left(g_j(\varphi_2) \wedge \bigwedge\limits_{l \leq k < j} g_k(\varphi_1) \right) \geq g_l(\varphi_2) = g_l(\varphi_1 \bigcup \varphi_2)$ Hence, (4) holds.

Then we prove that

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) \geq \bigvee_{j \geq l} \left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1) \right). \tag{5}$$

Suppose inequality (5) does not hold, then we have

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) < \bigvee_{j \geq l}\left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1)\right). \tag{6}$$

According to inequality (6), $\exists t \geq l$ such that

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) < g_t(\varphi_2) \wedge \bigwedge_{l \leq k < t} g_k(\varphi_1) \leq g_t(\varphi_2). \tag{7}$$

Next, we prove a recursive inequality

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) \geq g_{i+1}\left(\varphi_1 \bigcup \varphi_2\right). \tag{8}$$

If inequality (8) does not hold, then we have

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) < g_{i+1}\left(\varphi_1 \bigcup \varphi_2\right). \tag{9}$$

Since $g_i(\varphi_1 \bigcup \varphi_2) = g_i(\varphi_2) \vee (g_i(\varphi_1) \wedge g_{i+1}(\varphi_1 \bigcup \varphi_2))$, by substituting (9) into this equation, get

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) > g_i(\varphi_2) \vee (g_i(\varphi_1) \wedge g_i\left(\varphi_1 \bigcup \varphi_2\right)) = (g_i(\varphi_2) \vee g_i(\varphi_1)) \wedge (g_i(\varphi_2) \vee g_i\left(\varphi_1 \bigcup \varphi_2\right))$$

$> (g_i(\varphi_2) \vee g_i(\varphi_1)) \wedge g_i(\varphi_1 \bigcup \varphi_2)$. Therefore,

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) > g_i(\varphi_2) \vee g_i(\varphi_1) \tag{10}$$

However,

$g_i(\varphi_1 \bigcup \varphi_2) = g_i(\varphi_2) \vee (g_i(\varphi_1) \wedge g_{i+1}(\varphi_1 \bigcup \varphi_2))$
$= (g_i(\varphi_2) \vee g_i(\varphi_1)) \wedge (g_i(\varphi_2) \vee g_i(\varphi_1 \bigcup \varphi_2)) \leq g_i(\varphi_2) \vee (g_i(\varphi_1)$.

This contradicts inequality (10), so inequality (8) does not hold, and inequality (8) is proven.

From the arbitrariness of i in (8), it can be obtained that

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) \geq g_{i+1}\left(\varphi_1 \bigcup \varphi_2\right) \geq \cdots \geq g_t\left(\varphi_1 \bigcup \varphi_2\right). \tag{11}$$

$$\text{Since } g_t\left(\varphi_1 \bigcup \varphi_2\right) = g_t(\varphi_2) \vee (g_t(\varphi_1) \wedge g_{t+1}\left(\varphi_1 \bigcup \varphi_2\right) \geq g_t(\varphi_2), \tag{12}$$

by substituting inequalities (11) and (12) into (7), we have

$$g_l\left(\varphi_1 \bigcup \varphi_2\right) \geq g_t\left(\varphi_1 \bigcup \varphi_2\right) \geq g_l(\varphi_2) > g_l\left(\varphi_1 \bigcup \varphi_2\right).$$

This is a contradictory formula, so (6) does not hold, and therefore, (5) holds. Derived from inequalities (4) and (5), we have $g_l(\varphi_1 \bigcup \varphi_2) = \bigvee_{j \geq l}\left(g_j(\varphi_2) \wedge \bigvee_{l \leq k < j} g_k(\varphi_1)\right)$.

Then, by introducing the inductive hypothesis, we can obtain $g_l(\varphi_1 \bigcup \varphi_2)$ $= \bigvee_{j \geq l}\left(g_j(\varphi_2) \wedge \bigwedge_{l \leq k < j} g_k(\varphi_1)\right) = \left\|\pi^l, \varphi_1 \bigcup \varphi_2\right\|$. Lemma 2 has been proven. □

Lemma 3. $\forall i \in \mathbb{N}, g_i(\varphi_1 \bigcup \varphi_2) = \bigvee_{j \geq i}\left(g_j(\varphi_2) \wedge \bigwedge_{i \leq k < j} g_k(\varphi_1)\right) = \left\|\pi^i, \varphi_1 \bigcup \varphi_2\right\|$.

Proof. For all $i \in \mathbb{N}$, we only need to prove that

$$g_i\left(\varphi_1 \bigcup \varphi_2\right) = \bigvee_{j \geq i}\left(g_j(\varphi_2) \wedge \bigwedge_{i \leq k < j} g_k(\varphi_1)\right). \tag{13}$$

Now, using the first mathematical induction method to prove Equation (13), according to Lemma 2, Equation (13) holds when $i = l$. Now, assuming that Equation (13) holds, when $i = t$, then we have $g_t(\varphi_1 \bigcup \varphi_2) = \bigvee_{j \geq t}\left(g_j(\varphi_2) \wedge \bigwedge_{t \leq k < j} g_k(\varphi_1)\right)$.

We then need to show that it also holds for $i = t - 1$.

$$\begin{aligned}
&g_{t-1}(\varphi_1 \bigcup \varphi_2) = g_{t-1}(\varphi_2) \vee g_t(\varphi_1 \bigcup \varphi_2)\\
&= g_{t-1}(\varphi_2) \vee \bigvee_{j \geq t}\left(g_j(\varphi_2) \wedge \bigwedge_{t \leq k < j} g_k(\varphi_1)\right)(\textit{Induction Hypositesis})\\
&= (g_{t-1}(\varphi_2) \wedge 1) \vee \bigvee_{j \geq t}\left(g_j(\varphi_2) \wedge \bigwedge_{t \leq k < j} g_k(\varphi_1)\right)\\
&= (g_{t-1}(\varphi_2) \wedge \bigwedge_{t-1 \leq k < t-1} g_k(\varphi_1)) \vee \bigvee_{j \geq t}\left(g_j(\varphi_2) \wedge \bigwedge_{t \leq k < j} g_k(\varphi_1)\right)\\
&= \bigvee_{j \geq t-1}\left(g_j(\varphi_2) \wedge \bigwedge_{t-1 \leq k < j} g_k(\varphi_1)\right).
\end{aligned}$$

This is to say that, if Equation (13) holds when $i = t$, then, it can be proven that Equation (12) holds when $i = t - 1$. In addition, since Equation (13) holds when $i = l$ (as per Lemma 2), we can conclude by mathematical induction that Equation (13) holds $\forall t \in \{i, i+1, \cdots, t, \cdots, l\}$. From Lemma 2 (and the proof of Lemma 3), we obtain that

$$\left\|\pi^i, \varphi_1 \bigcup \varphi_2\right\| = g_i\left(\varphi_1 \bigcup \varphi_2\right). \square$$

Furthermore, it can be concluded that $\|\pi, \varphi_1 \bigcup \varphi_2\| = \left\|\pi^0, \varphi_1 \bigcup \varphi_2\right\| = g_0(\varphi_1 \bigcup \varphi_2) \in Q_0$. According to the definition of Q_0, we get $\|\pi, \varphi_1 \bigcup \varphi_2\| = g_0(\varphi_1 \bigcup \varphi_2) \in P$. We deduce that $\|\pi, \varphi\| \in P \Longleftarrow \pi \in L(A_{\varphi,P})$.

At this point, we have proven that $\|\pi, \varphi\| \in P$ if and only if $\pi \in L(A_{\varphi,P})$.

Finally, we will discuss the scale of $A_{\varphi,P}$, which was constructed in Theorem 3.

Let $V^+(\varphi) = \bigcup_{\phi \in cl(\varphi)} V(\phi) \in V(\varphi)$. Because $g : cl(\varphi) \to V^+(\varphi)$, so,

$$\begin{aligned}
|Q| \leq |g| = \left|V^+(\varphi)\right|^{cl(\varphi)} \leq |V(\varphi)|^{cl(\varphi)} = \left(m^{|\varphi|}\right)^{|\varphi|;}\\
= m^{|\varphi|^2;}|F_{\varphi_1 \bigcup \varphi_2}| \leq \left|g\left(\varphi_1 \bigcup \varphi_2\right)\right| \leq \left|cl(\varphi)\right| \leq \left|\varphi\right|.
\end{aligned}$$

At this point, the proof of Theorem 3 has been completed. □

4.2. The Validity and Satisfiability Issues of Boolean Reachability Semantics for QFLTL

Definition 13 [validity and satisfiability of the Boolean reachability semantics of QFLTL]. *Given a QFLTL formula φ, computing* $\boldsymbol{min}\left\{\|\pi, \varphi\| \middle| \pi \in \left([0,1]^{AP}\right)^{\omega}\right\}$ *is the problem of Boolean reachability semantic validity of QFLTL. Computing* $\boldsymbol{max}\left\{\|\pi, \varphi\| \middle| \pi \in \left([0,1]^{AP}\right)^{\omega}\right\}$ *is the problem of Boolean reachability semantic satisfiability of QFLTL.*

Solve the problem of Boolean reachability semantic validity of QFLTL through a divide and conquer strategy. The subproblem description for the decomposition of the Boolean reachable semantic validity problem is as follows.

Given a QFLTL formula φ, a fuzzy threshold $c \in [0,1]$, and $\forall \pi \in \left([0,1]^{AP}\right)^{\omega}$, determine whether $\|\pi, \varphi\| \geq \mathrm{c}$ holds or not.

$$\forall \pi \in \left([0,1]^{AP}\right)^{\omega}, \|\pi, \varphi\| \geq \mathrm{c} \Longleftrightarrow \neg\, \exists \pi \in \left([0,1]^{AP}\right)^{\omega}, \|\pi, \varphi\| \in [0, c) = P_{<c}.$$

Construct $A_{\varphi, P_{<c}}$ using the method of Theorem 3. If $\mathrm{L}\left(A_{\varphi, P_{<c}}\right) = \varnothing$, then $\forall \pi \in \left([0,1]^{\mathrm{AP}}\right)^{\omega}, \|\pi, \varnothing\| \geq c$ holds. Otherwise, it does not hold. If $\mathrm{L}\left(A_{\varphi, P_{<c}}\right) \neq \varnothing$, we can choose a smaller $c' < c$, and construct $A_{\varphi, P_{<c'}}$ using the method of Theorem 3. Then, it can easily be shown that $\mathrm{L}\left(A_{\varphi, P_{<c'}}\right) \subseteq \mathrm{L}\left(A_{\varphi, P_{<c}}\right)$. That is to say, $L\left(A_{\varphi, P_{<c}}\right)$ monotonically decreases as the threshold c decreases, in the sense of set inclusion order. In this way, we can utilize the binary search method to solve the Boolean reachable semantic validity problem.

The satisfiability problem for Boolean reachable semantics is described as follows. Given a QFLTL formula of φ and a fuzzy threshold of $c \in [0,1]$, $\exists \pi \in \left([0,1]^{AP}\right)^{\omega}$, such that $\|\pi, \varphi\| \geq \mathrm{c}$ holds or not, construct $A_{\varphi, P_{\geq c}}$ using the method of Theorem 3. If $L\left(A_{\varphi, P_{\geq c}}\right) \neq \varnothing$, then $\exists \pi \in \left([0,1]^{AP}\right)^{\omega}$, such that $\|\pi, \varphi\| \geq \mathrm{c}$ holds. Otherwise, it does not hold. Similar to the discussion above regarding the relationship with the threshold c, it is straightforward to conclude that $L\left(A_{\varphi, P_{\geq c}}\right)$ decreases as the threshold c increases, in the sense of set inclusion. In this way, we can also employ the idea of binary search to solve satisfaction problems.

Below, we will present algorithms for solving the validity and satisfiability problems for QFLTL with Boolean reachable semantics. These algorithms can provide solutions to the problems within a very small range of error ε that meets the requirements. The smaller the value of ε, the smaller the error of the solution, but the convergence speed of the algorithm will decrease. We will provide a rigorous proof process for this during the algorithm correctness proof and complexity analysis.

Algorithm 1 provides the validity problem-solving algorithm for QFLTL with Boolean reachable semantics. Algorithm 2 provides the satisfiability problem-solving algorithm for QFLTL with Boolean reachable semantics.

Theorem 4. *Given a QFLTL formula φ and an error threshold ε, let m be the cardinality of the measurable path set. Using Algorithms 1 and 2 to solve the QFLTL Boolean reachable semantic validity problem and satisfiability problem, respectively, is correct, and their complexity is $O\left(m^{|\varphi|^2} . \lceil log\left(\varepsilon^{-1}\right) \rceil\right)$.*

Proof. Let us first prove the correctness of the algorithms. The size of the set $L\left(A_{\varphi, P_{<mid[i]}}\right)$ is positively correlated with the length of the predicate interval $P_{<mid[i]} = [low[i],\ mid[i])$, where $i \in \mathbb{N}$. Based on this monotonicity, employ the binary search method to approximate the solution cv for the Boolean reachable semantic validity problem of QFLTL to a very small fuzzy interval $[low[k],\ high[k]]$, such that $cv = mid[k]$ and $l[k] = high[k] - low[k] \leq \varepsilon$. Here, ε is typically a given relatively small error threshold, representing the error between the final solution and the true solution. The smaller the value of ε, the closer the solution is to the true solution. But at the same time, the convergence speed of the algorithm will decrease. We will discuss the impact of ε on algorithm complexity later. The specific analysis is as follows.

If $L\left(A_{\varphi,P_{<mid[i]}}\right) = \emptyset$ in step 5 of the i-th iteration of Algorithm 1, it indicates that the range of the fuzzy predicate $P_{<mid[i]} = [low[i], mid[i]]$ is too narrow. Therefore let $high[i+1] = high[i]$ and $low[i+1] = mid[i]$. In step 3 of the $(i+1)$-th iteration,

$$mid[i+1] = (mid[i] + high[i])/2 > (low[i] + high[i])/2 = mid[i].$$

But note that the last equality is actually incorrect here because $mid[i+1]$ is defined as a new midpoint; the intended comparison must have been with $mid[i]$.

If $L\left(A_{\varphi,P_{<mid[i]}}\right) \neq \emptyset$ in step 8 of the i-th iteration of Algorithm 1, it indicates that the range of the fuzzy predicate $P_{<mid[i]} = [low[i], mid[i])$ is too wide. Therefore, let $high[i+1] = mid[i]$, $low[i+1] = low[i]$. For step 3 of the (i + 1)-th iteration,

$$mid[i+1] = (low[i] + mid[i])/2 < (low[i] + high[i])/2 = mid[i].$$

It is possible to narrow down the range of the fuzzy predicate to $P_{<mid[i]} = [low[i], mid[i])$.

By iteratively compressing the predicate $P_{<mid}$ until $P_{<mid[k]} = [low[k], mid[k])$ is compressed into an interval of length less than ε, we find the minimum $cv = mid[k]$ that ensures that $\mathbf{min}\left\{\|\pi, \varphi\| \middle| \pi \in \left([0,1]^{AP}\right)^{\omega}\right\} \in [cv - \varepsilon/2, cv + \varepsilon/2]$. In other words, when ε is used as an acceptable error threshold, we conclude that $cv = mid[k]$.

The correctness proof of Algorithm 2 is similar to that of Algorithm 1 and is, therefore, omitted here.

Now, let us analyze the algorithm complexity. The initial search interval length is $l[0] = 1 - 0 = 1$. The interval length corresponding to the i-th search is $l[i] = high[\mathrm{i}] - low[i]$, and the recursive relationship is $l[i+1] = l[i]/2$. A simple proof is as follows.

Because $mid[i] = low[i] \bigoplus high[i] = (low[i] + high[i])/2$;

If $low[i+1] = mid[i]$, then $high[i+1] = high[i]$, and $l[i+1]$ $= high[i+1] - low[i+1] = high[i] - (low[i] + high[i])/2 = (high[i] - low[i])/2 = l[i]/2$;

If $high[i+1] = mid[i]$, then $low[i+1] = low[i]$, and $l[i+1] = high[i+1] - low[i+1]$ $= (low[i] + high[i])/2 - low[i] = (high[i] - low[i])/2 = l[i]/2$;

So, we get $l[i+1] = l[i]/2$.

If the length of the final search interval is denoted as $l[k]$, then $l[k] = \varepsilon$. Thus, we have $\varepsilon = l[k] = l[0] \times (1/2)^k = (1/2)^k$, and solving for k gives $k = \log(\varepsilon^{-1})$. The i-th search requires constructing an automaton $A_{\varphi,P_{<mid[i]}}$ or $A_{\varphi,P_{>mid[i]}}$ with a complexity of $O\left(m^{|\varphi|^2}\right)$ (Refer to Theorem 3). Additionally, it is necessary to determine the null state of the automaton. The complexity of the automaton's null checking is linearly related to the size of the automaton [16]. Each binary search necessitates the construction of $A_{\varphi,P_{<mid[i]}}$, which is repeated a total of $k = \log(\varepsilon^{-1})$ times. So, the complexity of Algorithms 1 and 2 is $O\left(m^{|\varphi|^2} \cdot \lceil \log(\varepsilon^{-1}) \rceil\right)$. □

Obviously, the smaller the error threshold ε, the higher the algorithm complexity becomes. At the same time, the complexity of the algorithm grows exponentially as $|\varphi|^2$ increases. Fortunately, in reality, the formula length $|\varphi|$ is usually not very large, which indicates that Algorithm 1 and Algorithm 2 are effective.

4.3. Model Checking of QFLTL with Boolean Reachable Semantics

Model the system to obtain the physical symbol system M and represent the properties that the system satisfies using the temporal logic formula φ. Verify the satisfaction of φ on M through a model-checking algorithm. In classical model checking, if the checking result indicates that φ satisfies M, the checking process terminates. If φ does not satisfy M, return to the counterexample path in M for refinement and improvement. In quantitative model checking, the algorithm calculates the minimum satisfaction value of φ on M and outputs the path that achieves this minimum satisfaction value.

Algorithm 1. The solving algorithm for the Boolean reachable semantic validity problem of QFLTL.

Input: A QFLTL formula φ, an error threshold ε (for example $\varepsilon = 10^{-6}$);

Solution process:

1.Initialization: $low[0] = 0, high[0] = 1, i = 0$;
2 LOOP
3 $mid[i] = low[i] \oplus high[i]$;
4 Construct $A_{\varphi,P_{<mid[i]}}$ using the method of Theorem 3;
5 IF $L(A_{\varphi,P_{<mid[i]}}) = \emptyset$
6 $high[i+1] = high[\mathrm{i}], low[i+1] = mid[i]$;
7 END
8 ELSE $L(A_{\varphi,P_{<mid[i]}}) \neq \emptyset$
9 $high[i+1] = mid[i], low[i+1] = low[i]$;
10 END
11 $l[i] = high[\mathrm{i}] - low[i]$;
12 IF $l[i] \leq \varepsilon$
13 BREAK;
14 END
15 ELSE $l[i] > \varepsilon$
16 $i = i + 1$;
17 END
18 END
19 $cv = mid[i]$;

Output: cv.

Algorithm 2. The solving algorithm for Boolean reachable semantic satisfiability problem of QFLTL.

Input: A QFLTL formula φ, an error threshold ε (for example $\varepsilon = 10^{-6}$);

Solution process:

1. $Initialization$: $low[0] = 0, high[0] = 1, i = 0$;
2 LOOP
3 $mid[i] = low[i] \oplus high[i]$;
4 Construct $A_{\varphi,P_{\geq mid[i]}}$ using the method of Theorem 3;
5 IF $L(A_{\varphi,P_{\geq mid[i]}}) \neq \emptyset$
6 $high[i+1] = high[\mathrm{i}], low[i+1] = mid[i]$;
7 END
8 ELSE $L(A_{\varphi,P_{\geq mid[i]}}) = \emptyset$
9 $high[i+1] = mid[i], low[i+1] = low[i]$;
10 END
11 $l[i] = high[\mathrm{i}] - low[i]$;
12 IF $l[i] \leq \varepsilon$
13 BREAK;
14 END
15 ELSE $l[i] > \varepsilon$
16 $i = i + 1$;
17 END
18 END
19 $cs = mid[i]$;

Output: cs.

Definition 14 [Boolean reachable semantic model checking]. *Given a QFLTL formula φ and an FKS M, computing $argmin(\{\|\pi,\varphi\||\pi \in Path(M)\})$ is defined as the problem of performing Boolean reachable semantic model checking for QFLTL.*

That is, searching for the paths that start from the initial state in M, where the target paths are those with the minimum satisfaction value of φ among all such paths.

Theorem 5. *Let φ be a QFLTL formula, M be a dynamic transfer system, ε be a given error threshold, $mc = min(\{\|\pi,\varphi\||\pi \in Path(M)\})$. Furthermore, let $A_{\varphi,P_{<mc}}$, $A_{\varphi,P_{\leq mc}}$ and $A_{\varphi,P_{\geq mc}}$, be the NGBA constructed using the method of Theorem 3. Then the following conclusion is valid.*

(1) $Path\big(A_{\varphi,P_{<mc}} \otimes M\big) = \Phi$, $Path\big(A_{\varphi,P_{\leq mc}} \otimes M\big) \neq \Phi$;

(2) *Let* $mc = \|\pi,\varphi\|, \pi \in Path(M), A_{\varphi,P_{=mc}} = A_{\varphi,P_{\leq mc}} \otimes A_{\varphi,P_{\geq mc}}$, *then*

$$\pi \in L\big(A_{\varphi,P_{=mc}}\big) = L\Big(A_{\varphi,P_{\leq mc}} \bigotimes A_{\varphi,P_{\geq mc}}\Big);$$

(3) $\boldsymbol{argmin}(\{\|\pi,\varphi\||\pi \in Path(M)\}) = Path\Big(A_{\varphi,P_{=mc}} \otimes M\Big)$.

Proof. (1) First, we prove $\mathrm{Path}(\mathrm{A}_{\varphi,\mathrm{P}_{<\mathrm{mc}}} \otimes \mathrm{M}) = \emptyset$.

Given $mc = \mathbf{min}(\{\|\pi,\varphi\||\pi \in Path(M)\})$, it follows that

$$\forall\pi(\pi \in Path(M) \to \|\pi,\varphi\| \geq mc \Longleftrightarrow \neg\exists\pi(\pi \in Path(M) \wedge \|\pi,\varphi\| < mc).$$

Since mc is the minimum satisfaction value among all paths in M, there exists no path in M that has a satisfaction value of less than mc. Hence, $Path\big(A_{\varphi,P_{<mc}} \otimes M\big) = \emptyset$.

Now, we provide the proof for $Path\big(A_{\varphi,P_{\leq mc}} \otimes M\big) \neq \emptyset$.

Given $mc = \mathbf{min}(\{\|\pi,\varphi\||\pi \in Path(M)\})$, it implies that

$$\exists\pi(\pi \in Path(M) \wedge \|\pi,\varphi\| = mc).$$

Since there exists at least one path with a satisfaction value equal to mc, it also implies that $\exists\pi(\pi \in Path(M) \wedge \|\pi,\varphi\| \leq mc)$. Therefore, $Path\big(A_{\varphi,P_{\leq mc}} \otimes M\big) \neq \emptyset$;

(2) $mc = \|\pi,\varphi\|, \pi \in Path(M) \Longleftrightarrow \pi \in Path(M) \wedge \|\pi,\varphi\| \geq mc \wedge \|\pi,\varphi\| \leq mc$

$$\Longrightarrow (\pi \in Path(M) \wedge \|\pi,\varphi\| \geq mc) \wedge (\pi \in Path(M) \wedge \|\pi,\varphi\| \leq mc)$$
$$\Longleftrightarrow \pi \in L\big(A_{\varphi,P_{\leq mc}} \otimes A_{\varphi,P_{\geq mc}}\big) = L\big(A_{\varphi,P_{=mc}}\big)$$

(1) First, give the proof of $\mathbf{argmin}(\{\|\pi,\varphi\||\pi \in Path(M)\}) \subseteq Path\Big(A_{\varphi,P_{=mc}} \otimes M\Big)$.

$$\forall\pi_{mc} \in \mathbf{argmin}(\{\|\pi,\varphi\||\pi \in Path(M)\}) \Longrightarrow \pi_{mc} \in Path(M) \wedge \|\pi_{mc},\varphi\| = mc$$
$$\Longleftrightarrow \pi_{mc} \in Path(M) \wedge \pi_{mc} \in L\big(A_{\varphi,P_{=mc}}\big) \Longleftrightarrow \pi_{mc} \in Path\Big(A_{\varphi,P_{=mc}} \otimes M\Big).$$

Next, we provide the proof of $Path\Big(A_{\varphi,P_{=mc}} \otimes M\Big) \subseteq \mathbf{argmin}(\{\|\pi,\varphi\||\pi \in Path(M)\})$

$$\forall\pi_{mc} \in Path\Big(A_{\varphi,P_{=mc}} \otimes M\Big) \Longleftrightarrow \pi_{mc} \in L\big(A_{\varphi,P_{=mc}}\big) \wedge \pi_{mc} \in Path(M)$$
$$\Longleftrightarrow \Big(\pi_{mc} \in \Big([0,1]^{AP}\Big)^{\omega} \wedge \|\pi_{mc},\varphi\| = mc\Big) \wedge \pi_{mc} \in Path(M)$$
$$\Longrightarrow (\pi_{mc} \in Path(M) \wedge \|\pi_{mc},\varphi\| = mc) \wedge \pi_{mc} \in Path(M)$$
$$\Longleftrightarrow \pi_{mc} \in \{\pi|\pi \in Path(M), \|\pi,\varphi\| = mc\} \Longleftrightarrow \pi_{mc} \in \mathbf{argmin}(\{\|\pi,\varphi\||\pi \in Path(M)\}). \square$$

Under the constraint of the fuzzy predicate P, an automaton $A_{\varphi,P}$ is constructed based on the QFLTL (quantified fuzzy linear temporal logic) formula φ. The product of $A_{\varphi,P}$ and system model M yields $A_{\varphi,P} \otimes M$, which characterizes the satisfaction of system M with respect to the temporal property φ under the constraint of fuzzy predicate P. Based on this principle, a Boolean reachable semantic model-checking algorithm for QFLTL is proposed as Algorithm 3.

Algorithm 3. An automatic model-checking algorithm for Boolean reachable semantics in QFLTL.

Input: A QFLTL formula φ, an error threshold ε (for example $\varepsilon = 10^{-6}$), FKS M.

Solution process:

1. $Initialization$: $low[0] = 0, high[0] = 1, i = 0$;
2 LOOP
3 $mid[i] = low[i] \oplus high[i]$;
4 Construct $A_{\varphi,P_{\leq mid[i]}}$ using the method of Theorem 3;
5 Construct $A_{\varphi,P_{\leq mid[i]}} \otimes M$ using the method described in reference [17];
6 Use the method in reference [16] to determine the null of $A_{\varphi,P_{\leq mid[i]}} \otimes M$;
7 IF $A_{\varphi,P_{\leq mid[i]}} \otimes M = \emptyset$
8 $high[i+1] = high[\mathrm{i}], low[i+1] = mid[i]$;
9 END
10 ELSE $A_{\varphi,P_{\leq mid[i]}} \otimes M \neq \emptyset$;
11 $high[i+1] = mid[i], low[i+1] = low[i]$;
12 END
13 $l[i] = high[\mathrm{i}] - low[i]$;
14 IF $l[i] \leq \varepsilon$
15 BREAK;
16 END
17 ELSE $l[i] > \varepsilon$
18 $i = i + 1$;
19 END
20 END
21 $mc = mid[i]$
22 Construct $A_{\varphi,P_{\geq mid[i]}}$ using the method of Theorem 3;
23 Construct $A_{\varphi,P_{=mc}} = A_{\varphi,P_{\leq mc}} \otimes A_{\varphi,P_{\geq mc}}$;
24 Construct $A_{\varphi,P_{=mc}} \otimes M$;
25 Search for paths in $A_{\varphi,P_{=mc}} \otimes M$ and generate the path set $Path(A_{\varphi,P_{=mc}} \otimes M)$;

Output: $Path\left(A_{\varphi,P_{=mc}} \otimes M\right)$.

Theorem 6. *Given a QFLTL formula φ and an error threshold ε; Let $M = (S, I, \delta, AP, L)$ be an FKS. Using Algorithm 3 to solve the QFLTL Boolean reachable semantic model-checking problem is correct. The complexity of the algorithm is* $O\left(|S|^{|\varphi|^2} \cdot \left(\lceil log\left(\varepsilon^{-1}\right)\rceil + |S|^{|\varphi|^2+1}\right)\right)$.

Proof. According to Definition 14, the model-checking problem aims to find the paths in M that correspond to the minimum satisfaction value of φ. Steps 2–20 of Algorithm 3 employ a binary search method to continuously narrow the search interval until the satisfied value of φ in M is compressed to an interval where $|\|M, \varphi\| - mc| < \varepsilon$. Thus, $mc = \mathbf{min}(\{\|\pi, \varphi\| | \pi \in Path(M)\})$ is obtained. In steps 22–24 of the algorithm, based on Theorems 3 and 5, it can be deduced that

$$\mathbf{argmin}(\{\|\pi, \varphi\| | \pi \in Path(M)\}) = Path\left(A_{\varphi,P_{=mc}} \bigotimes M\right).$$

Now, we analyze the complexity of Algorithm 3. Based on Theorem 3, the complexity of constructing an automaton is $O\left(m^{|\varphi|^2}\right)$, where m is the cardinality of the measurable paths. In the model-checking problem, the maximum cardinality of the path is the number of states $|S|$. Therefore, the complexity of constructing the automaton $A_{\varphi,P}$ is $\mathrm{O}\left(|S|^{|\varphi|^2}\right)$.The number of states for $A_{\varphi,P} \otimes M$ is $\mathrm{O}\left(\left|A_{\varphi,P}\right| \cdot |S|\right)$. So, the complexity of constructing $A_{\varphi,P} \otimes M$ is $O\left(|S|^{|\varphi|^2+1}\right)$. The complexity of the null detection algorithm for $A_{\varphi,P} \otimes M$ is linearly related to its size [16,17], which is $O\left(|S|^{|\varphi|^2+1}\right)$. The complexity of computing mc is $\mathrm{O}\left(|S|^{|\varphi|^2+1} \cdot \left(\lceil \log\left(\varepsilon^{-1}\right)\rceil\right)\right)$ (refer to Theorem 4). The size of the product

of two NGBAs is the product of their sizes [18]. The size of $A_{\varphi,P_{=mc}} = A_{\varphi,P_{\leq mc}} \bigotimes A_{\varphi,P_{\geq mc}}$ is $O(|A_{\varphi,P_{=mc}}|) = O(|A_{\varphi,P_{\leq mc}}|) \times O\left(|A_{\varphi,P_{\geq mc}}|\right) = O\left(|S|^{2|\varphi|^2}\right)$. The size of $A_{\varphi,P} \bigotimes M$ is $O(|A_{\varphi,P}|) \times O(|S|) = O\left(|S|^{2|\varphi|^2} \times |S|\right) = O\left(|S|^{2|\varphi|^2+1}\right)$. The complexity of searching for paths in $A_{\varphi,P} \bigotimes M$ is linearly related to its size. Hence, the overall complexity of model checking is $O\left(|S|^{|\varphi|^2}.\left(\lceil \log(\varepsilon^{-1}) \rceil + |S|^{|\varphi|^2+1}\right)\right)$. □

5. Search and Decision Problems for Quantitative Reachable Semantic in QFLTL

Quantitative reachability semantics integrate path-reachability information with formula satisfaction value information, where path reachability is defined on a specified path fragment. The search and decision problem-solving for temporal logic is typically grounded in automata construction and emptiness checking or via the composition of fuzzy matrices. However, for QFLTL, neither of these approaches suffices. First, automata construction for temporal logic formulas solely considers the values satisfied by the formulas and fails to incorporate quantitative transition metrics between states. Second, as information composition is no longer constrained to conjunction ("∧"), it becomes impractical to compute the quantitative reachable semantics of QFLTL formulas through fuzzy matrix composition operations. Nonetheless, a straightforward approach involves searching for paths in finite Kripke structures (FKSs) and subsequently calculating reachable semantics along these paths. First, in practical system operation, the limited application of sequence, selection, and loop rules during system transitions indeed encompasses primarily lasso paths. Second, an FKS comprising solely of lasso paths can induce a finite tree structure, thereby reducing the search and decision problems of QFLTL quantitative reachability semantics to those based on this spanning tree. Consequently, we will restrict the paths in the FKS to either pure lasso paths or lasso paths exclusively. Subsequently, we offer solutions to search and decision-making problems within these specialized FKSs.

Let $M = (S, I, \delta, AP, L)$ be an FKS. We introduce the concept of relationships between states as follows. $\forall s \in S$, $\text{Child}(s) = \{s' | s' \in S, \delta(s, s') > 0\}$ is the set of subsequent states of s, also known as the set of child nodes of s.

For all $i \in \mathbb{N}, \text{Child}^i(s) \in \begin{cases} \{s\} & i = 0; \\ \text{Child}(\text{Child}^{i-1}(s)) & i > 0. \end{cases}$ is called the i-th generation descendant node of s.

For all $s \in S$, $\text{Fater}(s) = \{s' | \text{s} \in \text{Child}(s')\}$ is the set of prefix states for s, also known as the set of parent nodes for s.

For all $i \in \mathbb{N}, \text{Father}^i(s) \in \begin{cases} \{s\} & i = 0; \\ \text{Father}(\text{Father}^{i-1}(s)) & i > 0. \end{cases}$ is called the i-th generation ancestor node of s.

For all $n \in \mathbb{N}^+$ (where $\mathbb{N}^+$ represents a non-zero set of natural numbers), $I_n = \{0, 1, \cdots, n-1\}$ is called an n-related ordinal set.

5.1. Search and Decision Problems on Pure Lasso FKSs

Definition 15 [pure lasso FKSs] *Given an FKS of $M = (S, I, \delta, AP, L)$, a pure lasso FKS accompanied with M is a tuple $M_{PL} = (M, \Pi_{PL})$, where*

$$\Pi_{PL} = \{\pi | \pi \in Path(M), \pi \text{ is a pure lasso path.}\}.$$

To solve the search and decision problems on pure lasso FKSs, we transform them into an equivalent finite tree structure when computing the QFLTL formula value.

Definition 16 [spanning tree of pure lasso FKSs]. *Let $M_{PL} = (M, \Pi_{PL})$ be a pure lasso FKS. The spanning tree T_M_{PL} of M_{PL} is defined as follows.*

(1) $\forall s \in M$ in M_{PL}, if $I(s) > 0$ then s is the root node of T_M_{PL};

(2) *If s is a node of T_M_{PL}, $\forall s' \in Child(s)$ in M, then add node s' to the spanning tree as a child of s and add an edge (s, s') with a weight of $\delta(s, s')$. If $\forall i \in \mathbb{N}^+$, it holds that $s' \neq Father^i(s')$. Then, expand s' recursively. Otherwise, s' is a leaf node, and the extension of this branch ends.*

According to Definition 16, the nodes of a pure lasso FKS can be continuously expanded without generating a loop, and a corresponding spanning tree for the pure lasso FKS can be constructed. Algorithm 4 outlines the procedure for generating the spanning tree of a pure lasso FKS.

Algorithm 4. The algorithm for generating spanning trees of a pure lasso FKS.

Input: A pure lasso FKS $M_{PL} = (M, \Pi_{PL})$, a table OPEN, and a table CLOSED.
//The OPEN table is used to store nodes to be extended, while the CLOSED table stores the spanning tree of T_M_{PL}.

Solution process:

```
1 LOOP ∀s ∈ M_PL
2     IF I(s) > 0
3         Add s to table OPEN;//s is the root node of T_M_PL.
4     END
5 END
6 LOOP OPEN is not empty
7     Remove the first node s_n from the OPEN table, and add it to the CLOSED table;
8     B_Node = s_n//Mark the node for tracing back.
9     LOOP Father(B_Node) ≠ NULL
10        s_f = Father(B_Node);
11        IF s_f = s_n//The current extension node is a duplicate of its immediate ancestor.
12            Child(s_n) = NULL //Extension of node s_n ended.
13            BREAK;
14        END
15        ELSE s_f ≠ s_n;
16            B_Node = s_f;
17        END
18    END
19    IF Father(B_Node) = NULL//The current extension node does not have a duplicate direct
          ancestor node.
20        LOOP ∀s ∈ Child(s_n)//Traverse the child nodes of s
21            Add s to OPEN table;
22            Father(s) = s_n;//Establishing a backtracking pointer.
23        END
24    END
25 END
```

Output: table CLOSED.

Definition 17. *$M_{PL} = (M, \Pi_{PL})$ is a pure lasso FKS, and T_M_{PL} is the spanning tree of M_{PL}. The sequence of nodes $\pi_{fin} = s_0, s_1, \cdots, s_{n-1}, s_n$, from the root node to a leaf node in T_M_{PL} is called a path of T_M_{PL}. $Path(T\_M_{PL})$ is the set of paths in T_M_{PL}.*

Theorem 7 *If $\pi_{fin} = s_0, s_1, \cdots, s_{n-1}, s_n$ is a path in T_M_{PL}, and φ is a QFLTL formula. Then,*

(1) $s_n \in \{s_0, s_1, \cdots, s_{n-1}\}$, $B\left(\pi_{fin}\right) = \{s_0, s_1, \cdots, s_{n-1}\}$, $\left|\pi_{fin}\right| = n \leq |S|$;

(2) *The computational complexity of $[\![\pi_{fin}, \varphi]\!]$ is $O\left((|S|+3)^{|\varphi|}\right)$.*

Proof. According to Definitions 11, 12, 16, and 17, the result of (1) in theorem 7 holds. Let $t\left([\![\pi_{fin}, \varphi]\!]\right)$ denote the number of computations performed for π_{fin} with respect to φ. When selecting the operators"$\circ$" and "$\bigoplus_\lambda$", the maximum number of operations is three. When selecting "λ_{ne}" or "λ_{cf}" for "$\bigtriangleup_1$", two operations are performed. Computing $\delta^i_*(\pi_{fin})$ requires i times the "$\bigwedge$"operation. Now, we summarize and prove (2) based on the structure of φ.

(1) When $\varphi = r$, $[\![\pi_{fin}, \varphi]\!] = r \circ \delta_*^i\left(\pi_{fin}\right)$; $t\left([\![\pi_{fin}, \varphi]\!]\right) = i + 3 \le |S| + 3 \le \left(|S| + 3\right)^{|\varphi|}$;

(2) When $\varphi = p$, $[\![\pi_{fin}, \varphi]\!] = L\left(\pi_{fin}^i\right)(p) \circ \delta_*^i\left(\pi_{fin}\right)$; $t\left([\![\pi_{fin}, p]\!]\right) = i + 3 \le \left(|S| + 3\right)^{|\varphi|}$;

(3) When $\varphi = \bigtriangleup_1 \phi$, $[\![\pi_{fin}, \bigtriangleup_1 \phi]\!] = \bigtriangleup_1\left([\![\pi_{fin}, \phi]\!]\right), t\left([\![\pi_{fin}, \bigtriangleup_1 \phi]\!]\right) = t\left([\![\pi_{fin}, \phi]\!]\right) + 2$. According to the inductive assumption,

$$t([\![\pi_{fin}, \bigtriangleup_1 \phi]\!]) =\le (|s| + 3)^{|\phi|} + 2 < (|s| + 3)^{|\phi|+1} = (|s| + 3)^{|\varphi|};$$

(4) When $\varphi = \varphi_1 \bigtriangleup_2 \varphi_2$, $[\![\pi_{fin}, \varphi_1 \bigtriangleup_2 \varphi_2]\!] = [\![\pi_{fin}, \varphi_1]\!] \bigtriangleup_2 [\![\pi_{fin}, \varphi_2]\!]$

$$\begin{aligned} t\left([\![\pi_{fin}, \varphi]\!]\right) &= t\left([\![\pi_{fin}, \varphi_1]\!]\right) + t\left([\![\pi_{fin}, \varphi_2]\!]\right) + 3 \\ &\le (|S| + 3)^{|\varphi_1|} + (|S| + 3)^{|\varphi_2|} + 3 < (|s| + 3)^{|\varphi_1|+|\varphi_2+1} = (|S| + 3)^{|\varphi|} \end{aligned}$$

(5) When $\varphi = \bigcirc\phi$, $[\![\pi_{fin}, \bigcirc\phi]\!] = [\![\pi_{fin}^1, \phi]\!]$. According to the inductive assumption,

$$t([\![\pi_{fin}, \bigcirc\phi]\!]) = t([\![\pi_{fin}^1, \phi]\!]) \le (|S| + 3)^{|\phi|} < (|S| + 3)^{|\varphi|};$$

(6) when $\varphi = \Diamond\phi$, $[\![\pi_{fin}, \Diamond\phi]\!] = \bigvee\limits_{0 \le j < n} [\![\pi_{fin}^j, \phi]\!]$;

$$\begin{aligned} &t([\![\pi_{fin} \Diamond\phi]\!]) = n \cdot t([\![\pi_{fin}^i, \phi]\!]) \le n \cdot (|S| + 3)^{|\phi|} \\ &\le |S| \cdot (|S| + 3)^{|\phi|} < (|S| + 3)^{|\phi|+1} = (|S| + 3)^{|\varphi|}; \end{aligned}$$

(7) When $\varphi = \Box\phi$,

$$[\![\pi_{fin}, \phi]\!] = \bigwedge_{0 \le j < i} [\![\pi_{fin}^i, \phi]\!],\ t\left([\![\pi_{fin}, \Box\phi]\!]\right) = n \cdot t\left([\![\pi_{fin}^i, \phi]\!]\right) \le \left(|S| + 3\right)^{|\varphi|};$$

(8) When $\varphi = \varphi_1 \bigcup \varphi_2$, $[\![\pi_{fin}, \varphi_1 \bigcup \varphi_2]\!] = \bigvee\limits_{0 \le i < n} \left([\![\pi_{fin}, \varphi_2]\!] \wedge \bigwedge\limits_{0 \le j < i} [\![\pi_{fin}, \varphi_1]\!]\right)$

$$\begin{aligned} t([\![\pi_{fin}, \varphi_1 \bigcup \varphi_2]\!]) &\le n \cdot (t([\![\pi_{fin}^i, \varphi_2]\!]) + \mathrm{t}([\![\pi_{fin}^j, \varphi_2]\!]) + \mathrm{n}) \\ &\le |S| \cdot (t([\![\pi_{fin}^i, \varphi_2]\!]) + \mathrm{t}([\![\pi_{fin}^j, \varphi_2]\!])) + |S|^2 \le |S| \end{aligned}$$

$$\begin{aligned} &((|S| + 3)^{|\varphi_1|} + (|S| + 3)^{|\varphi_2|}) + |S|^2 < |S| \cdot ((|S| + 3)^{|\varphi_1|} + (|S| + 3)^{|\varphi_2|} + (|S| + 3)) \\ &< (|S| + 3) \cdot ((|S| + 3)^{|\varphi_1|} + (|S| + 3)^{|\varphi_2|} + (|S| + 3)). \end{aligned}$$

Let $|S| + 3 = k \in \mathrm{N}, k > 3\ \forall k_1, k_2 \in N^+$. Now, we prove that $k^{k_1} + k^{k_2} + k < k^{k_1+k_2}$. This equation is equivalent to $1/k^{k_2} + 1/k^{k_1} + 1/k^{k_1+k_2-1} < 1$. Because $1/k^{k_2} \le 1/4, 1/k^{k_1} \le 1/4, 1/k^{k_1+k_2-1} \le 1/7$, $1/k^{k_2} + 1/k^{k_1} + 1/k^{k_1+k_2-1} < 1$ is correct, and $k^{k_1} + k^{k_2} + k < k^{k_1+k_2}$ is also proven. In this case,

$$\begin{aligned} &t([\![\pi_{fin}, \varphi_1 \cup \varphi_2]\!]) < (|S| + 3) \cdot ((|S| + 3)^{|\varphi_1|} + (|S| + 3)^{|\varphi_2|} + (|S| + 3)) \\ &< (|S| + 3) \cdot (|S| + 3)^{|\varphi_1|+|\varphi_2|} = (|S| + 3)^{|\varphi_1|+|\varphi_2|+1} = (|S| + 3)^{|\varphi|}. \square \end{aligned}$$

Corollary 1. *$M_{PL} = (M, \Pi_{PL})$ is a pure lasso FKS, and T_M_{PL} is the spanning tree of M_{PL}. $\forall \pi \in Path(M_{PL})$. There exists a unique $\pi_{fin} \in Path(T\_M_{PL})$, so that $[\![\pi, \varphi]\!] = [\![\pi_{fin}, \varphi]\!]$.*

Proof. According to Definition 17 and Algorithm 4, $\forall \pi \in Path(M_{PL})$, π has the form $\pi = \pi_0, \pi_1, \cdots, \pi_{u-1}, (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega$, and there exists a unique, $\pi_{fin} \in Path(T_{MPL})$, $\pi_{fin} = \pi_0, \pi_1, \cdots, \pi_{u-1}, \pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1}$, such that,

$$\exists \pi_{u+v} \in \{s | s \in Child(\pi_{u+v-1})\}, \text{and } \pi_{u+v} = \pi_u.$$

The uniqueness here is guaranteed by the tree structure of T_M_{PL}, where exists is a unique path from the root node to the leaf node in the tree. It is straightforward to prove $[\![\pi, \varphi]\!] = [\![\pi_{fin}, \varphi]\!]$ by induction on the structure φ. Here is an example to illustrate the situation about $\varphi = \varphi_1 \bigcup \varphi_2$.

Let $H_i = [\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge\limits_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]$, where $i \in \mathbb{N}$ and

$$\pi = \pi_0, \pi_1, \cdots, \pi_{u-1}, (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega.$$

We have $H_{u+v} = H_\mu$ for all $\pi_i \in (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega, \exists k \in \mathbb{N},\ j \in \{u, u+1, \cdots u+v-1\}$, such that $\pi_i = \pi_{j+kv} = \pi_j$.

In this way, for all $i \ge u,\ \exists j \in \{u, u+1, \cdots, u+v-1\}$, such that $H_i = H_j$. Therefore,

$$\begin{aligned}
&[\![\pi, \varphi_1 \bigcup \varphi_2]\!] = \bigvee_{i\ge 0}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right)\\
&= \bigvee_{0\le i\le u-1}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right) \vee \bigvee_{u\le i\le u+v-1}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right)\\
&\vee \bigvee_{u+v\le i}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right)\\
&= \bigvee_{0\le i\le u-1}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right) \vee \bigvee_{u\le i\le u+v-1}\left([\![\pi^i,\ \varphi_2]\!] \wedge \bigwedge_{0\le j<i} [\![\pi^j,\ \varphi_1]\!]\right)\\
&= [\![\pi_{fin}, \varphi_1 \cup \varphi_2]\!].\square
\end{aligned}$$

Next, we provide a formal description and a solution algorithm for search and decision problems in pure lasso FKSs.

Definition 18 [quantitative reachable semantic validity and satisfiability in pure lasso FKSs]. *Let $M_{PL} = (M, \Pi_{PL})$ be a pure lasso FKS, and T_M_{PL} be the spanning tree of M_{PL}. The problem of calculating $\boldsymbol{min}\{[\![\pi_{fin}, \varphi]\!] \big| \pi_{fin} \in Path(T\_M_{PL})\}$ is defined as the quantitative reachable semantic validity problem on the pure lasso FKS. Similarly, the problem of calculating $\boldsymbol{max}\{[\![\pi_{fin}, \varphi]\!] \big| \pi_{fin} \in Path(T\_M_{PL})\}$ is defined as the problem of quantitatively reachable semantic satisfiability on the pure lasso FKS.*

Algorithm 5 is the algorithm for solving the quantitative reachable semantic validity and satisfiability problems in QFLTL on pure lasso FKSs.

Algorithm 5. Quantitative reachable semantic validity and satisfiability problem-solving algorithm in QFLTL on pure lasso FKSs.

Input: A pure lasso FKS $M_{PL} = (M, \Pi_{PL})$, and a QFLTL formula φ.

Solution process:
1 Calculate the spanning tree T_M_{PL} of M_{PL} according to Algorithm 4;
2 Perform a depth −
first search on T_M_{PL}, when a leaf node is encountered, backtrack to generate a path π_{fin} from the root node to that leaf node;
3 For the path π_{fin} generated in step 2 $\left(\text{where } \pi_{fin} \in Path(T\_M_{PL})\right)$, use the "Challenge Arena Algorithm" to find the maximum and minimum values : Let $mvp = \mathbf{min}\{[\![\pi_{fin}, \varphi]\!] \big| \pi_{fin} \in Path(T\_M_{PL})\}$. Let $msp = \mathbf{max}\{[\![\pi_{fin}, \varphi]\!] \big| \pi_{fin} \in Path(T\_M_{PL})\}$.

Output: mvp, msp.

Theorem 8. *Given a QFLTL formula φ and a pure lasso FKS, in Algorithm 5, mvp and msp are the validity and satisfiability values of φ on M_{PL}. The complexity of the algorithm is $O\left(|S|!\cdot\left|S\right|^{|\varphi|}\right)$.*

Proof. According to Theorem 7 and Corollary 1, the output results of Algorithm 5, *mvp*, and *msp*, are the validity and satisfiability values of φ on $\mathrm{M_{PL}}$. According to Algorithm 4, solve the spanning tree T_{MPL} of M_{PL}. There can be a maximum of $|S|$ root nodes in the tree. Each root node can have up to $|S|-1$ children, and the first layer can have up to $|S|-1$ branches. The nodes in the i-th layer have i direct ancestors, with a maximum of $|S|-i$ children, and can generate $|S|-i$ branches. Hence, T_{MPL} has a maximum of $|S|\times(|S|-1)\times(|S|-2)\times\cdots\times(|S|-(|S|-1))=|S|!$ leaf nodes. Each leaf node corresponds to path π_{fin}, with a length not exceeding $|S|$ from the root node, so T_{MPL} has a maximum of $\mathrm{O}(|S|!)$ paths. According to Theorem 7, evaluating φ on π_{fin} has a complexity of $O\left(\left(\left|S\right|+3\right)^{|\varphi|}\right)$ on π_{fin}. The complexity of the overall algorithm is the product of the number of paths and the complexity of evaluating φ on π_{fin}, so the complexity of Algorithm 5 is $\mathrm{O}\left(|S|!\cdot\left|S\right|^{|\varphi|}\right)$. In practice, however, due to the fact that the number of subsequent states in T_{MPL} is often less than $|S|$, the number of paths is generally much smaller than $O(|S|!)$. Therefore, the average complexity of the algorithm is significantly lower than $O\left(|S|!\cdot\left|S\right|^{|\varphi|}\right)$. □

5.2. Searching and Decision-Making Problems on Lasso FKSs

Next, we relax the constraints of the system production rules and consider the searching and decision-making problems on systems containing general lasso paths. The partitioning of system paths essentially depends on the constraints and partitioning of rules generated by the system. We provide the following formal description for systems that contain only lasso paths.

Definition 19 [lasso FKSs]. *Given an FKS $M=(S,I,\delta,AP,L)$, the lasso FKS accompanied by M is a tuple $M_L=(M,\Pi_L)$, where $\Pi_L=\{\pi|\pi\in Path(M),\pi$ is a lasso path.$\}$.*

Definition 20 [division about generating rules]. *Let M be an FKS. The production rules are defined as follows.*

(1) Sequential transition rule set, $\delta_s=\{\delta(s,s')|\forall s\in S,s'\in Child(s),|Child(s)|=1\}$;
(2) Selective transition rule set, $\delta_a=\{\delta(s,s')|\forall s\in S,s'\in Child(s),|Child(s)|>1\}$;
(3) Finite cycle transition rule set,

$$\delta_{l^+}=\left\{\left(\delta(s_0,s_1),\delta(s_1,s_2),\cdots,\delta(s_{n-1},s_n)\right)^+\middle|\forall i\in I_n,\delta(s_i,s_{i+1})\in\ \delta_s,\ and\ s_0=s_n\right\};$$

(4) Infinite loop transition rule set,

$$\delta_{l^\omega}=\{(\delta(s_0,s_1),\delta(s_1,s_2),\cdots,\delta(s_{n-1},s_n))^\omega|\forall i\in I_n,\delta(s_i,s_{i+1})\in\ \delta_s,\ and\ s_0=s_n\}.$$

Theorem 9 [decision theorem about lasso FKSs]. *If M is an FKS, then M_L is a lasso FKS associated with M, if and only if the transition function of M_L is defined as follows.*

$\delta_L=\{\delta_s,\delta_a,\delta_{l^+},\delta_{l^\omega}|\delta_s,\delta_a,\delta_{l^+},\delta_{l^\omega}$ arefinitesets.$\}$.

Proof. First, we prove the necessity. Let M_L be the lasso FKS accompanied by FKS M. From Definitions 19 and 20, the elements of M_L are known to be lasso paths that have formed $\pi=\pi_0,\pi_1,\cdots,\pi_{u-1},(\pi_u,\pi_{u+1},\cdots,\pi_{u+v-1})^\omega$. On this form of path, a finite path, the finite fragments $\pi_0,\pi_1,\cdots,\pi_{u-1}$ and states $\pi_u,\pi_{u+1},\cdots,\pi_{u+v-1}$ appear in sequence, which are generated by a finite number of sequential rules. It allows for the occurrence of a

finite number of loop fragments in the sequence, which are generated by the finite number of uses of finite loop transition rules. That is,

$$\forall i \in I_{u+v}\delta(\pi_i, \pi_{+1}) \in \delta_s \cup \delta_{l^+}, |\delta_s \cup \delta_{l^+}| < \infty, \ (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega \in \delta_{l^\omega}.$$

$\forall s \in S$ if $|C\mathit{hild}(s)| \geq 2$. There will be $|C\mathit{hild}(s)|$ finite branches starting from s in M, corresponding to one use of selection generation rules in M_L, resulting in Child(s) lasso paths. Hence, the transition function of M_L is defined as

$$\delta_L = \{\delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} | \delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} \text{ are finite sets.}\}.$$

For the proof of sufficiency, the transition function of M_L is δ_L. Using the select transition rules once will generate a finite number of branches, with each lasso path corresponding to one branch. The constraint of a lasso path pertains only to one path. Thus, we only need to consider the application of rules for sequence generation, finite loop generation, and infinite loop generation on a single branch. Applying δ_s, δ_a, and δ_{l^+} a finite number of times within δ_L can generate a path segment with a finite number of loop fragments, denoted as $\pi_0, \pi_1, \cdots, \pi_{u-1}, \pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1}$. Since the path in M is infinite, infinite loops must be generated using the infinite loop transition rule δ_{l^ω}. We use δ_{l^ω} exactly once. According to the definition of δ_{l^ω}, let us assume that, in the path segment $\pi_0, \pi_1, \cdots, \pi_{u-1}, \pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1}$, we have $(\delta(\pi_u, \pi_{u+1}), \cdots, \delta(\pi_{u+v-2}, \pi_{u+v-1}))^\omega$. This ultimately produces an infinite path, with the form $\pi = \pi_0, \pi_1, \cdots, \pi_{u-1}, (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega$. According to Definition 11, π is a lasso path. In this manner, we obtain the spanning FKS M_L of M, and the transition function of M_L is $\delta_L \subseteq \delta$. We prove that the path generated by δ_L are lasso paths. That is to say, by restricting the transition function of M to δ_L, we can obtain the lasso FKS M_L that is accompanied by M. □

Corollary 2 [decision theorem about pure lasso FKSs]. *If M is an FKS, then M_{PL} is the pure lasso FKS associated with M if and only if the transition function of M_{PL} is,*

$$\delta_{PL} = \{\delta_s, \delta_a, \delta_{l^\omega}, |\delta_s, \delta_a, \delta_{l^\omega} \text{ are finite sets. }\}.$$

Pure lasso (Definition 12) requires an infinite loop to appear on the path, and this infinite loop is generated using the infinite loop transition rule δ_{l^ω} exactly once. There are no duplicate states on the path segments outside and inside the loop, which is why the rules for generating pure lasso FKSs do not include the finite cycle transition rule. The proof of Corollary 1 is similar to that of Theorem 9, focusing on the necessary and sufficient conditions for M_{PL} to be a pure lasso FKS associated with M.

Definition 21 [quantitative reachable semantic validity and satisfiability in lasso FKSs]. *Given a QFLTL formula of φ and an FKS, $M = (S, I, \delta, AP, L)$, where $M_L = (M, \Pi_L)$ is the lasso FKS associated with M. Computing $\boldsymbol{min}\{[\![\pi_{fin}, \varphi]\!] \Big| \pi_{fin} \in Path(T\_M_L)\Big\}$ is defined as the quantitative reachable semantic validity problem in lasso FKSs, and computing $\boldsymbol{max}\{[\![\pi_{fin}, \varphi]\!] \Big| \pi_{fin} \in Path(T\_M_L)\Big\}$ is defined as the problem of quantitatively reachable semantic satisfiability problem in lasso FKSs.*

Theorem 10. *Given a QFLTL formula of φ and an FKS of M, let $M_L = (M, \Pi_L)$ be the lasso FKS associated with M. There exists a pure lasso FKS M_{PL} that is associated with M_L, such that*

(1) $\boldsymbol{min}\{[\![\pi, \varphi]\!] | \pi \in Path(M_L)\} = \boldsymbol{min}\{[\![\pi, \varphi]\!] | \pi \in Path(M_{PL})\}$;
(2) $\boldsymbol{max}\{[\![\pi, \varphi]\!] | \pi \in Path(M_L)\} = \boldsymbol{max}\{[\![\pi, \varphi]\!] | \pi \in Path(M_{PL})\}$.

Proof. According to Definitions 11 12, 16, and 19, Theorem 9 and Corollary 2, a pure lasso path is a special case of a lasso path. A finite number of finite loop fragments can appear on the inner and outer path fragments of an infinite loop segment in a lasso path,

while a pure lasso path does not contain such finite loop fragments. The proof of Theorem 10 involves treating $k \in \mathbb{N}^+$ finite loop fragments on the lasso path in M_L as infinite loop fragments and then decompose the original lasso path to generate $k+1$ pure lasso paths. We prove that the validity and satisfiability problem on the original lasso path are equivalent to the validity and satisfiability problems on the newly generated set of $k+1$ pure lasso paths. Let $\pi = \pi_0, \pi_1, \cdots, \pi_{u-1}, (\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega$ be a lasso path in M_L, where $\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1}$ is the path fragment where the infinite loop is located, and $\pi_0, \pi_1, \cdots, \pi_{u-1}$, is the path fragment outside the infinite loop. Suppose a finite loop segment $(\pi_a, \pi_{a+1}, \cdots, \pi_{a+m-1})^+ \subseteq \pi_0, \pi_1, \cdots, \pi_{u-1}$ appears within $\pi_0, \pi_1, \cdots, \pi_{u-1}$. We construct a pure lasso path as follows.

$$\pi' = \pi_0, \pi_1, \cdots, \pi_{a-1}, \pi_a, \pi_{a+1}, \cdots, \pi_{a+m-1}, \cdots, \pi_{u-1}(\pi_u, \pi_{u+1}, \cdots, \pi_{u+v-1})^\omega$$

In this path, the states in the finite loop fragment only appear in sequence once. Then, construct another pure lasso path, where $\pi'' = \pi_0, \pi_1, \cdots, \pi_{a-1}, (\pi_a, \pi_{a+1}, \cdots, \pi_{a+m-1})^\omega$. Once the state in π enters a finite loop fragment, a pure lasso path π'' is generated based on this segment using the infinite repetition generation rule. When $(\pi_a, \pi_{a+1}, \cdots, \pi_{a+m-1})^+ \subseteq \pi_0, \pi_1, \cdots, \pi_{u-1}$, construct two pure lasso paths. One path is π' . The states in the finite loop fragment of π' only appear sequentially once in each infinite loop. Another path is constructed as π''. Once the state in π enters a finite loop segment, a pure lasso path π'' is generated based on this fragment using the infinite repetition generation rule. It can be summarized according to the structure of φ, and it is straightforward to prove that

$$\text{when calculating } \mathbf{min}\{\, [\![\pi, \varphi]\!] | \pi \in Path(M_L)\,\}, \text{ we have } [\![\pi, \varphi]\!] = [\![\pi', \varphi]\!] \wedge [\![\pi'', \varphi]\!]. \quad (14)$$

$$\text{When calculating } \mathbf{max}\{\, [\![\pi, \varphi]\!] | \pi \in Path(M_L)\,\}, \text{ we have } [\![\pi, \varphi]\!] = [\![\pi', \varphi]\!] \vee [\![\pi'', \varphi]\!]. \quad (15)$$

When there are $k > 1$ finite loop fragments within the lasso path π, a similar approach can be applied. The finite loop fragment traverses the state of all finite loop fragments outside the infinite loop of π only once. If the finite loop fragment is within the infinite loop of π, each time it enters the infinite loop, it traverses the state of the finite loop segment once. In this scenario, we generate a pure lasso $\pi(0)$. Traverse the state of the *i*-1 loop segment before the *i-th* finite loop segment only once, and after entering the *i-th* finite loop segment, use the loop fragment infinitely to generate a pure lasso path. In this way, a finite pure lasso path cluster $\{\pi(i) | i \in I_k\}$ accompanied by the lasso path π is constructed. Similar to Equations (14) and (15),

$$\text{when calculating } \mathbf{min}\{\, [\![\pi, \varphi]\!] | \pi \in Path(M_L)\,\}, \text{ we have } [\![\pi, \varphi]\!] = \bigwedge_{0 \le j < k} [\![\pi(i), \varphi]\!]. \quad (16)$$

$$\text{When calculating } \mathbf{max}\{\, [\![\pi, \varphi]\!] | \pi \in Path(M_L)\,\}, \text{ we have } [\![\pi, \varphi]\!] = \bigvee_{0 \le j < k} [\![\pi(i), \varphi]\!]. \quad (17)$$

Obviously, according to Algorithm 4, when generating the pure lasso-spanning tree T_M_{PL} on M_L, it includes all pure lasso paths in the pure lasso path cluster $\{\pi(i) | i \in I_k\}$ that are accompanied by the lasso path π. Consequently, the conclusion of Theorem 10 holds. □

In order to solve the quantitative reachable semantic validity and satisfiability problem of QFLTL on lasso FKSs. It is possible to construct a pure lasso FKS that accompanies the original lasso FKS during the proof process, based on Theorem 10. Subsequently, the quantitative semantic validity and satisfiability problem-solving algorithm of QFLTL on the pure lasso FKS is used to solve the original problem.

Algorithm 6. Algorithm for solving the quantitative reachable semantic validity and satisfiability problem for QFLTL on lasso FKSs.

Input: Given a QFLTL formula φ and a Lasso FKS $M_L = (M, \prod_L)$.
//The production rule for M_L is denoted as $\delta_L = \{\delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} | \delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega}$ are finite sets.$\}$.

Solution process:
1 According to Algorithm 4, construct the spanning tree T_M_L of M_L;
2 Perform a depth-first search on T_M_L. If a leaf node is encountered, backtracking generates a path π_{fin} from the root node to that leaf node;
3 For each path $\left\{ \pi_{fin} \middle| \pi_{fin} \in Path(T\_M_{PL}) \right\}$ generated in step 2, use the "Challenge Arena Algorithm" to find the maximum and minimum values:
$mv = \mathbf{min}\{ [\![\pi_{fin}, \varphi]\!] \middle| \pi_{fin} \in Path(T\_M_{PL}) \}$; $ms = \mathbf{max}\left\{ [\![\pi_{fin}, \varphi]\!] \middle| \pi_{fin} \in Path(T\_M_{PL}) \right\}$.

Output: mv, ms.

Theorem 11 *Given a QFLTL formula φ and a lasso FKS M_L, in Algorithm 6, mv and ms are the validity and satisfiability values of φ on M_L. The complexity of the algorithm is $O\left(|S|! \cdot \left|S\right|^{|\varphi|}\right)$.*

The proof of Theorem 11 is analogous to the proof of Theorem 8.

5.3. Quantitative Reachable Semantic Model Checking for QFLTL

A quantitative model-checking algorithm for QFLTL should return the path set within the system that achieves the minimum value of the given temporal logic formula. formula.

Definition 22 [quantitative reachable semantic model checking]. *Given a QFLTL formula of φ and an FKS of M, let $M_L = (M, \Pi_L)$ be the lasso FKS associated with M, and $M_{PL} = (M, \Pi_{PL})$ be the lasso FKS associated with M_L. Computing $\boldsymbol{arg}(\boldsymbol{min}\{[\![\pi, \varphi]\!] | \pi \in Path(M_{PL})\})$ is defined as the quantitative reachable semantic model-checking problem in pure lasso FKSs. Computing $\boldsymbol{arg}(\boldsymbol{min}\{ [\![\pi, \varphi]\!] | \pi \in Path(M_L) \})$ is defined as the quantitative reachable semantic model-checking problem in lasso FKSs.*

The paths in a pure lasso FKS correspond one-to-one with those in its spanning tree. In the spanning tree, a depth-first search is performed to find a leaf node, and then, backtracking is employed from the leaf node to the root node to generate one path after another. Each generated path is evaluated, and the "Challenge Arena Algorithm" is utilized to record the set of paths that achieve the minimum value of the given QFLTL formula. Below is a model-checking algorithm for QFLTL on pure lasso FKSs, based on Algorithm 4.

Algorithm 7. The model-checking algorithm for QFLTL on pure lasso FKSs.

Input: A QFLTL formula φ, an FKS M, and pure lasso FKS M_{PL} that associated with M.
//The production rule for M_{PL} is $\delta_{PL} = \{\delta_s, \delta_a, \delta_{l^\omega}, | \delta_s, \delta_a, \delta_{l^\omega}$ are finite sets. $\}$.

Solution process:
1 Initialize $\Pi_{mvp} = \emptyset$;
2 According to Algorithm 4, construct the spanning tree T_M_{PL} of M_{PL};
3 LOOP Perform depth-first search on T_M_{PL}, if a leaf node is encountered, backtrack to generate a path π_{fin} from the root node to that leaf node;
4 IF $\Pi_{mvp} = \emptyset$ THEN $\Pi_{mvp} = \{\pi_{fin}\}$;
5 ELSE *take any* $\pi'_{fin} \in \Pi_{mvp}$;
6 IF $[\![\pi_{fin}, \varphi]\!] = [\![\pi'_{fin}, \varphi]\!]$ THEN $\Pi_{mvp} = \Pi_{mvp} \cup \{\pi_{fin}\}$;
7 IF $[\![\pi_{fin}, \varphi]\!] < [\![\pi'_{fin}, \varphi]\!]$ THEN $\Pi_{mvp} = \{\pi_{fin}\}$;
8 END

Output: Π_{mvp}.

Next, we will provide a model-checking algorithm for QFLTL on lasso FKSs.

Algorithm 8. Model-checking algorithm for QFLTL on lasso FKSs.

Input: QFLTL formula φ, FKS $M = (S, I, \delta, AP, L)$, lasso FKS M_L associated with M, table OPEN, and table CLOSED.
//The production rule for M_L is $\delta_L = \{\delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} | \delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} are\ finite\ set\}$. The OPEN table is used to store node will be extended, and the CLOSED table is used to store M_L.

Solution process:
1 Initialize $\Pi_{mvl} = \emptyset$;
2 $\forall s \in M_L$ IF $I(s) > 0$ THEN add s to the OPEN table;
3 LOOP the OPEN table is not empty
4 Remove the first node s from the OPEN table and add it to the CLOSED table;
5 IF s exists , extended s based on the production $(\delta(s_0, s_1), \delta(s_1, s_2), \cdots, \delta(s_{n-2}, s_{n-1}))^\omega$.
6 Add $s_1, s_2, \cdots, s_{n-1}$ to Table CLOSED one by one;
7 $\forall i \in I_n$, set $Father(s_{i+1}) = s_i$;
8 Backtrack from s_{n-1} along the Father pointer to a certain state s_{in} such that $I(s_{in}) > 0$, forming a finite path $\pi_{fin} = s_{in}, \cdots, s_1, s_2, \cdots, s_{n-1}$;
9 IF $\Pi_{mvl} = \emptyset$ THEN $\Pi_{mvl} = \{\pi_{fin}\}$;
10 ELSE $for\ any\ \pi'_{fin} \in \Pi_{mvp}$
11 IF $[\![\pi_{fin}, \varphi]\!] = [\![\pi'_{fin}, \varphi]\!]$ THEN $\Pi_{mvl} = \Pi_{mvl} \cup \{\pi_{fin}\}$;
12 IF $[\![\pi_{fin}, \varphi]\!] < [\![\pi'_{fin}, \varphi]\!]$ THEN $\Pi_{mvl} = \{\pi_{fin}\}$;
13 END
14 IF s exists , extended s based on the production $(\delta(s_0, s_1), \delta(s_1, s_2), \cdots, \delta(s_{n-2}, s_{n-1}))^+$.
15 Add $s_1, s_2, \cdots, s_{n-1}$ to table CLOSED one by one;
16 $\forall i \in I_n$, set $Father(s_{i+1}) = s_i$; Add s_{n-1} to table OPEN;
17 END
18 IF s exists , extended s based on the production $\delta(s_0, s_1), \delta(s_1, s_2), \cdots, \delta(s_{n-2}, s_{n-1}))$.
19 Add $s_1, s_2, \cdots, s_{n-1}$ to Table CLOSED one by one;
20 $\forall i \in I_n$, set $Father(s_{i+1}) = s_i$, Add s_{n-1} to table OPEN;
21 END
22 IF s exists, extended s based on the production $\{\delta(s, s') | s' \in Child(s), k = |Child(s)| > 1\}$
23 Add $s_1, s_2, \cdots, s_{n-1}$ to Table CLOSED one by one;
24 $\forall i \in I_k - \{0\}$, establish a Child pointer from s to s_i;
25 END
26 END

Output: Π_{mvl}.

Theorem 12 *Given a QFLTL formula of φ and an FKS of M, let $M_L = (M, \Pi_L)$ be the lasso FKS associated with M, and let $M_{PL} = (M, \Pi_{PL})$ be the pure lasso FKS associated with M. If Π_{mvp} and Π_{mvl} are the output results of Algorithms 7 and 8, respectively, then,*

(1) $\Pi_{mvp} = \boldsymbol{arg}(\boldsymbol{min}\{[\![\pi, \varphi]\!] | \pi \in Path(M_{PL})\})$, *and the complexity of Algorithm 7 is* $O(|S|! \cdot (|S| + 3)^{|\varphi|})$;

(2) $\Pi_{mvl} = \boldsymbol{arg}(\boldsymbol{min}\{[\![\pi, \varphi]\!] | \pi \in Path(M_L)\})$, *and the complexity of Algorithm 8 is* $O(|\delta_{l^\omega}| \cdot (|S| + 3)^{|\varphi|})$.

Proof. According to Theorem 9 and Corollary 2, the CLOSED tables constructed by Algorithms 7 and 8 correspond to M_{PL} and M_L, respectively. The proof process, which is similar to that of Algorithm 4, generates the $O(|\delta_{l^\omega}|)$ and O(|S|!) paths during the dynamic generation of M_{PL} and M_L, respectively. According to Theorem 7, the complexity of computing $[\![\pi_{fin}, \varphi]\!]$ on a finite path π_{fin} is $O((|S| + 3)^{|\varphi|})$. Therefore, the conclusion of Theorem 12 is correct. □

6. Examples of QFLTL Reasoning

Next, we present an example of using FKS to model the patient treatment process, employing QFLTL to characterize expectations for the treatment plan and solve related search and decision-making problems. Figure 3 depicts an FKS M that has been obtained by modeling the patient's treatment process.

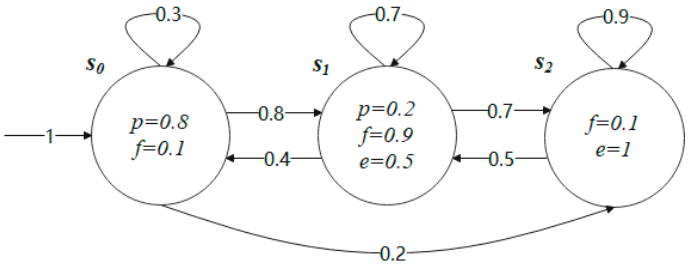

Figure 3. An FKS M Representing the patient treatment process.

The atomic proposition set $AP = (p, e, f)$; " p, e, f", respectively, indicate that the patient's health status is "poor", "fine", and "excellent". The state set S represents the patient's physical state during the treatment process. $I = (1, 0, 0)$ indicates that the initial state of the patient is determined as state s_0. δ is transition function. $\delta(s_0, s_0) = 0.3$ represents that the possibility of the patient remaining in state s_0 is 0.3. $\delta(s_0, s_1) = 0.8$ represents a probability of 0.8 for the patient to transition from state s_0 to s_1 after treatment.$\delta(s_0, s_2) = 0.2$ indicates that the patient has a 0.2 possibility of directly transferring from state s_0 to s_2 after treatment. $\delta(s_1, s_0) = 0.4$ indicates that the patient will deteriorate from state state s_1 to s_0 with a possibility of 0.4. Other transitions are similar. L is a label function. $L(\mathrm{s}_0)(p) = 0.8$. The possibility of "poor" status when the patient is in state s_0 is 0.8. $L(\mathrm{s}_0)(f) = 0.1$. The possibility of the health condition being "fine" when the patient is in state s_0 is 0.1. $L(\mathrm{s}_0)(e) = 0$. The possibility of the health condition being "excellent" when the patient is in state s_0 is zero. Other label values are similar. Different paths correspond to different treatment processes. For example, $\pi\_0 = s_0^{\omega}$ indicates that the patient has been in state s_0 continuously. $\pi\_1 = s_0, s_1^{\omega}$ indicates that the patient has been changed from state s_0 to s_1 after treatment and then remains in state s_1. $\pi\_2 = s_0^7, s_1^{\omega}$ indicates that the patient has changed from state s_0 to s_1 after one week of treatment in state s_0 and has been in state s_1 continuously. $\pi\_3 = s_0^5, s_1^7, s_2^{\omega}$ indicates that the patient has changed from state s_0 to state s_1 after 5 days of treatment in state s_0. After another 7 days of treatment, the state changed from s_1 to s_2 and then remained in state s_2 $\pi\_4 = s_0^7, s_2^{\omega}$, indicating that the patient has changed from state s_0 to s_2 after one week of treatment in state s_0 and has been in state s_2 continuously. The explanation of other paths will not be elaborated here.

Here are some examples of computing related to path reachability.

$$\delta_i^*(\pi_0) = \delta_i^*(\pi_1) = \delta_i^*(\pi_2) = \delta_i^*(\pi_3) = \begin{cases} 1 & i = 0; \\ 0.3 & i > 0. \end{cases} \delta_i^*(\pi\_4) = \begin{cases} 1 & i = 0; \\ 0.3 & 0 < i \leq 7; \\ 0.2 & i > 7. \end{cases}$$

There are nine pure lasso paths in M, which are

$$\pi\_pl\_0 = s_0^{\omega}; \pi\_pl\_1 = (s_0, s_1)^{\omega}; \ \pi\_pl\_2 = s_0, s_1^{\omega}; \pi\_pl\_3 = s_0, (s_1, s_2)^{\omega};$$
$$\pi\_pl\_4 = s_0, s_1, s_2^{\omega}; \pi\_pl\_5 = (s_0, s_2, s_1)^{\omega}; \ \pi\_pl\_6 = s_0, s_2, {s_1}^{\omega};$$
$$\pi\_pl\_7 = s_0, (s_2, s_1)^{\omega}; \text{ and } \pi\_pl\_8 = s_0, s_2^{\omega}.$$

There are 11 forms of lasso paths in M, which are

$$\pi\_l\_0 = s_0^{\omega}; \pi\_l\_1 = (s_0^+, s_1^+)^{\omega}; \pi\_l\_2 = s_0^+, s_1^{\omega}; \ \pi\_l\_3 = s_0^+, (s_1^+, s_2^+)^{\omega};$$
$$\pi\_l\_4 = s_0^+, s_1^+, s_2^{\omega}; \pi\_l\_5 = (s_0, s_1)^+, s_2^{\omega}; \ \pi\_l\_6 = (s_0^+, s_2^+, s_1^+)^{\omega};$$
$$\pi\_l\_7 = (s_0^+, (s_2, s_1)^+)^{\omega}; \ \pi\_l\_8 = s_0^+, s_2^+, s_1^{\omega};$$
$$\pi\_l\_9 = s_0^+, (s_2^+, s_1^+)^{\omega}; \text{ and } \pi\_l\_10 = s_0^+, s_2^{\omega}.$$

The pure lasso FKS accompanied by M is $M_{PL} = (M, \Pi_{PL})$, where

$$\Pi_{PL} = \{\pi\_pl\_i | i \in I_9\}.$$

The lasso FKS accompanied by M is $M_L = (M, \Pi_L)$, where $\Pi_L = \{\pi\_l\_i | i \in I_{11}\}$.

The calculation of the reachability for the pure lasso path is as follows.

$$\delta_i^*(\pi\_pl\_0) = \begin{cases} 1 & i = 0; \\ 0.3 & i > 0. \end{cases} \delta_i^*(\pi\_pl\_1) = \begin{cases} 1 & i = 0; \\ 0.8 & i = 1; \\ 0.4 & i > 1. \end{cases} \delta_i^*(\pi\_pl\_2) = \begin{cases} 1 & i = 0; \\ 0.8 & i = 1; \\ 0.7 & i > 1. \end{cases} \ldots\ldots$$

Some properties characterized by the QFLTL formula are given as follows.

$\varphi_1 = \Diamond\left(0.8_{cf}(e)\right)$. This formula is used to represent the possibility that the final physical condition of a treated patient is considered "excellent", with a maximum possibility of 0.8.

$\varphi_2 = \Box(f \bigoplus_{0.4} e)$. This formula is used to represent the possibility that the patient's physical condition consistently satisfies the criteria of 'fine' with a weight of 40% and 'excellent' with a weight of 60% during the treatment process.

$\varphi_3 = 0.5_{ne}(f) \bigcup e$. This formula represents the possibility that, throughout the treatment process, the patient's physical condition will consistently remain at a level considered 'fine' with a minimum necessity of at least 0.5 until it transitions to an 'excellent' level.

It is evident that the FLTL formula cannot express these properties, and neither can the LTL formula.

Some examples of Boolean reachable semantic paths are as follows.

$$\begin{aligned} &\|\pi\_pl\_0,\ \varphi_1\| = \bigvee_{i\geq 0} \|(\pi\_pl\_0)^i,\ 0.8_{cf}(e)\| = 0.8 \times \bigvee_{i\geq 0} 0 = 0 \\ &\|\pi\_pl\_1,\ \varphi_1\| = 0.8 \times (0 \vee 0.5 \vee 0 \vee 0.5 \vee \cdots \vee 0 \vee 0.5) = 0.4 \\ &\|\pi\_pl\_2,\ \varphi_1\| = \bigvee_{i\geq 0} \|(\pi\_pl\_2)^i,\ 0.8_{cf}(e)\| = 0.8 \times (0 \vee 0.5 \vee \cdots \vee 0.5) = 0.4; \end{aligned}$$

$\forall k \in I_9, k > 2, \|\pi\_pl\_k,\ \varphi_1\| = 0.8$. The reason is that the state s_2 occurs on all of these paths, and $s_2(e) = 1$, which represents the maximum possible value for e. Taking into account the quality constraint operator $0.8_{cf}(x)$ acting on e, the satisfaction value is 0.8.

$$\begin{aligned} &\forall k \in I_9,\ \ \|\pi\_pl\_k,\ \varphi_2\| = \bigwedge_{i\geq 0} (0.4 \times \|(\pi\_pl\_k)^i,\ f\| + 0.6 \times \|(\pi\_pl\_k)^i,\ e\| = 0.04. \\ &\|\pi\_pl\_0,\ \varphi_3\| = \bigvee_{i\geq 0} (\|(\pi\_pl\_0)^i,\ e\| \wedge \bigwedge_{0\leq j<i} \|(\pi\_pl\_0)^j,\ 0.5_{ne}(f)\|) = \bigvee_{i\geq 0} (0 \wedge \\ &\bigwedge_{0\leq j<i} \|(\pi\_pl\_0)^j,\ 0.55\|) = 0. \end{aligned}$$

We have listed the Boolean reachability semantics of φ_1, φ_2, and φ_3 on different lasso paths in Table 2 below for convenient comparative analysis.

Table 2. The Boolean reachability satisfaction values of $\boldsymbol{\varphi_1}$, $\boldsymbol{\varphi_2}$, and $\boldsymbol{\varphi_3}$ on different lasso paths.

$\|\pi\_l\_i, \varphi_j \mid \mid\|$	π_pl_0	π_pl_1	π_pl_2	π_pl_3	π_pl_4	π_pl_5	π_pl_6	π_pl_7	π_pl_8	$\left\lvert\{\left\|\pi_{l_i}, \varphi_j\right\|\}\right\rvert$
φ_1	0	0.4	0.4	0.8	0.8	0.8	0.8	0.8	0.8	3
φ_2	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	1
φ_3	0	0.5	0.5	0.55	0.55	0.55	0.55	0.55	0.55	1

In quantitative reachability calculations, the weighted average operator "$\oplus_{0.3}$" is selected as the composition operation "$\circ$". For all $\pi \in \left([0,1]^{AP}\right)^{\omega}, \varphi \in AP \cup APR, i \in \mathbb{N}$, according to Definition 6, we have $[\![\pi^i, \varphi]\!] = 0.3\delta_i^*(\pi) + 0.7[\![\pi_i, \varphi]\!]$. This indicates that the path reachability contributes 30% to the overall satisfaction value of formula φ, while the

property satisfaction value contributes 70% to the overall satisfaction value of formula φ. Below are some examples of such calculations.

$$\begin{aligned}
&[\![\pi_l_0,\ \varphi_1]\!] = \bigvee_{i\geq 0} [\![(\pi_l_0)^i, 0.8_{cf}(e)]\!] = \bigvee_{i\geq 0} \delta_i^*(\pi_l_0) \oplus_{0.3} 0.8(\pi_l_0)_i(e)\\
&= \bigvee_{i\geq 0} 0.3 \times \delta_i^*(\pi_l_0) + 0.7 \times 0.8 \times (\pi_l_0)_i(e)\\
&= \bigvee_{i\geq 0} 0.3\times\delta_i^*(\pi_l_0) + 0.56 \times 0) = 0.3 \times (\bigvee_{i\geq 0} \delta_i^*(\pi_{l_0})) = 0.3 \times (1 \vee \bigvee_{i\geq 1} 0.3) = 0.3;\\
&[\![\pi_l_0,\ \varphi_2]\!] = \bigwedge_{i\geq 0} [\![(\pi_l_0)^i,\ f \oplus_{0.4} e]\!]\\
&= \bigwedge_{i\geq 0}(0.3 \times \delta_i^*(\pi_{l_0}) + 0.7 \times (0.4 \times (\pi_l_0)_i(f) + 0.6 \times (\pi_l_0)_i(e)))\\
&= \bigwedge_{i\geq 0} (0.3 \times \delta_i^*(\pi_{l_0}) + 0.28 \times (\pi_l_0)_i(f) + 0.42 \times (\pi_l_0)_i(e)))\\
&= \bigwedge_{i\geq 0} (0.3 \times \delta_i^*(\pi_{l_0}) + 0.28 \times (\pi_l_0)_i(f))\\
&= (0.3 \times 1 + 0.28 \times 0.1) \wedge \bigwedge_{i\geq 1} (0.3 \times 0.3 + 0.28 \times 0.1) = 0.118.\\
&[\![\pi_l_0,\ \varphi_3]\!] = \bigvee_{i\geq 0} ([\![(\pi_l_1)^i, e\ ]\!] \wedge \bigwedge_{0\leq j<i} [\![(\pi_l_1)^j, 0.5_{ne}(f)\ ]\!]\\
&= \bigvee_{i\geq 0} ((0.3 \times \delta_i^*(\pi_l_0) + 0.7 \times (\pi_l_0)_i(e)) \wedge \bigwedge_{0\leq j<i} (0.3 \times \delta_j^*(\pi_l_0) + 0.7 \times (0.5\\
&\qquad +0.5 \times (\pi_l_0)_j(f)))\\
&= \bigvee_{i\geq 0} (0.3 \times \delta_i^*(\pi_l_0) + 0.7 \times 0) \wedge \bigwedge_{0\leq j<i} (0.3 \times \delta_j^*(\pi_l_0) + 0.35 + 0.35 \times 0.1)\\
&= \bigvee_{i\geq 0} (0.3 \times \delta_i^*(\pi_l_0) \wedge \bigwedge_{0\leq j<i} (0.3 \times \delta_j^*(\pi_l_0) + 0.385))\\
&= (0.3 \times 1 \vee \bigvee_{i\geq 1} (0.3 \times 0.3 \wedge \bigwedge_{0\leq j<i} (0.3 \times 0.3 + 0.385)) = 0.3.
\end{aligned}$$

We have listed the quantitative reachability satisfaction values of φ_1, φ_2, and φ_3 on different lasso paths in Table 3 below for convenient comparative analysis.

Table 3. The quantitative reachability satisfaction values of $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$, and $\boldsymbol{\varphi}_3$ on different lasso paths.

$[\![\pi\_l\_i, \varphi_j]\!]$	π_pl_0	π_pl_1	π_pl_2	π_pl_3	π_pl_4	π_pl_5	π_pl_6	π_pl_7	π_pl_8	$\lvert\{[\![\pi_{l_i}, \varphi_j]\!]\}\rvert$
φ_1	0.3	0.52	0.52	0.77	0.77	0.77	0.62	0.62	0.62	4
φ_2	0.118	0.148	0.328	0.328	0.328	0.088	0.328	0.328	0.328	4
φ_3	0.3	0.59	0.59	0.625	0.625	0.625	0.685	0.685	0.685	4

In the literature [7,8], the possibility of temporal logic is studied, which solely utilizes the classical max–min operation for information synthesis, excluding quality constraint operators such as a, b, and c. As a result, it cannot express the more nuanced properties like φ_1, φ_2, and φ_3 in QFCLTL. We have computed the satisfaction values of the PoLTL formulas $\phi_1 = \diamond e$, $\phi_2 = \Box(f \wedge e)$, and $\phi_3 = f \bigcup e$, which correspond to these three QFLTL formulas, on the KFS in the example regarding the patient treatment process. which correspond to these three QFLTL formulas, on the KFS in the example regarding the patient treatment process. The calculation results are presented in Table 4 to facilitate comparison and analysis between QFLTL and PoLTL.

Table 4. The Satisfaction Values of $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$, and $\boldsymbol{\phi}_3$ on different paths.

$[\![\pi_{l_i}]\!](\phi_j)$	π_pl_0	π_pl_1	π_pl_2	π_pl_3	π_pl_4	π_pl_5	π_pl_6	π_pl_7	π_pl_8	$\lvert\{[\![\pi_{l_i}]\!](\phi_j)\}\rvert$
ϕ_1	0	0.5	0.5	**0.7**	**0.7**	**0.7**	**0.2**	**0.2**	**0.2**	4
ϕ_2	0	0	0	0	0	0	0	0	0	1
ϕ_3	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	2

After conducting a comparative analysis of Tables 2–4, we can draw the following conclusions.

(1) QFLTL exhibits a stronger expressive power compared to PoLTL. First, PoLTL is incapable of expressing temporal properties that incorporate qualitative constraints, such as φ_1, φ_2, and φ_3, which specify particular qualities. Second, QFLTL allows for a more precise and quantitative characterization of system properties, as demonstrated by the last column in the three tables, which represents the diverse number of satisfaction values attained by the corresponding formulas within the system. It is clear that, within the same system, QFLTL formulas differentiate between values with greater granularity than those of PoLTL formulas;

(2) PoLTL may result in information loss, whereas QFLTL does not suffer from this limitation. PoLTL solely utilizes the "$\wedge$" operation to combine information between the path-reachability and property formulas, ultimately selecting the smaller of the two values as the satisfaction measure. For instance, in Table 4, the bold values indicate the path reachability, whereas the non-bold values correspond to the values of the property formulas. In contrast, the QFLTL in Table 3 consistently synthesizes information from both aspects;

(3) PoLTL fails to differentiate between the importance of sub-formulas within a property formula, nor does it make a distinction between the significance of the property formula itself and path reachability. In contrast, QFLTL adeptly discriminates between these types of information. For example, both the PoLTL formula $\phi_2 = \Box(f \wedge e)$ and the QFLTL formula $\varphi_2 = \Box(f \bigoplus_{0.4} e)$ necessitate that f and e occur concurrently along the path. Nonetheless, ϕ_2 selects the smaller value between f and e, whereas φ_2 integrates f and e with a weight ratio of 0.4:0.6, suggesting that e is regarded as the more essential property. During the synthesis of path reachability and property satisfaction values, PoLTL merely chooses the smaller value, whereas the QFLTL formulas can blend the two using the $\bigoplus_{0.3}$ operation, applying a weight ratio of 0.3:0.7 and indicating that QFLTL considers the property formula to be more important than path reachability;

(4) PoLTL may cause a desynchronization between the values of the property formula and the path-reachability information, whereas QFLTL can always ensure that these two pieces of information remain synchronized. For example, on the path $\pi\_pl\_3 = s_0, s_1, s_2, s_1, s_2, \cdots$, ..., the value of ϕ_3 is 0.1 for all path segments after the first occurrence of s_1, which actually reflects the value of $s_0(f) = 0.1$. However, in reality, the reachability values for all path segments after the second occurrence of s_1 are 0.5. On the contrary, φ_3, which uses the $\bigoplus_{0.3}$ operation to combine the current path reachability with the value of the property formula, successfully synchronizes these two pieces of information and results in four distinct values (as detailed in the fourth row of Table 3). In contrast, ϕ_3 only exhibits two values, namely 0 and 0.1.

Given an error threshold of ε = 0.001, here are examples to illustrate how algorithms one to eight can be used to obtain solutions for related searching and decision problems in QFLTL.

Using Algorithms 1–3 within FKS M, the individual problems of validity, satisfiability, and model checking, each under Boolean reachability semantics, were solved for the three QFLTL formulas φ_1, φ_2, and φ_3. The computational results of these problems are summarized in the following Table 5.

Table 5. Results for solving QFLTL Boolean reachability semantics-related problems within M.

$\boldsymbol{\varphi_i}$	**Algorithm 1 Validity** $\mathbf{min}\{\|\|\pi, \varphi_i\|\|\pi \in Path(M)\}$	**Algorithm 2 Satisfiability** $\mathbf{max}\{\|\|\pi, \varphi_i\|\|\pi \in Path(M)\}$	**Algorithm 3 Model Checking** $\mathbf{arg\ min}(\{\|\|\pi, \varphi_i\|\|\pi \in Path(M)\})$
φ_1	0.000	0.800	$\{s_0^\omega\}$
φ_2	0.040	0.040	$\{s_0^\omega\}$
φ_3	0.000	0.550	$\{s_0^\omega\}$

Based on the "Decision Theorem about Pure Lasso FKSs"(Corollary 2), by limiting the generation rule of KFS M to $\delta_{PL} = \{\delta_s, \delta_a, \delta_{l^\omega}, |\delta_s, \delta_a, \delta_{l^\omega}\ are\ finite\ sets\}$, we obtain its accompanying pure lasso FKS M_{PL}.

Generate the spanning tree T_M_{PL} of M_{PL} using Algorithm 4, as depicted in Figure 4, where the dashed edge "- - - - - -▶" represents an infinite loop of path fragments starting from the same direct ancestor node as the leaf node that the edge reaches along its path. The paths in T_M_{PL} are pure lasso paths.

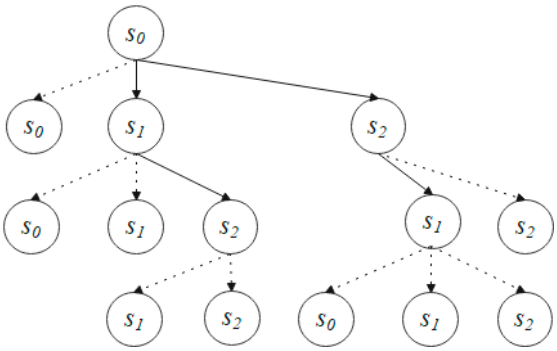

Figure 4. T_M_{PL}.

For example, Figure 5 shows a pure lasso path $\pi\_pl\_3 = s_0, (s_1, s_2)^\omega$.

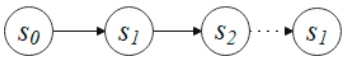

Figure 5. A pure lasso path $\pi\_pl\_3 = s_0, (s_1, s_2)^\omega$.

Using Algorithms 5 and 7 within M_{PL}, the individual problems of validity, satisfiability, and model checking, each under Boolean reachability semantics, were solved for the three QFLTL formulas φ_1, φ_2, and φ_3. The computational results of these problems are summarized in the following Table 6.

Table 6. Results for solving QFLTL quantitative reachability semantics-related problems within M_{PL}.

$\boldsymbol{\varphi_i}$	**Algorithm 5 Validity** $\mathbf{min}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_{PL})\}$	**Algorithm 5 Satisfiability** $\mathbf{max}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_{PL})\}$	**Algorithm 7 Model Checking** $\mathbf{arg(min}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_{PL})\})$
φ_1	0.300	0.770	$\{s_0^\omega\}$
φ_2	0.118	0.088	$\{s_0^\omega\}$
φ_3	0.300	0.685	$\{s_0^\omega\}$

Based on the "Decision Theorem about Lasso FKSs"(Theorem 9), by limiting the generation rule of KFS M to $\delta_L = \{\delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega} | \delta_s, \delta_a, \delta_{l^+}, \delta_{l^\omega}$ *are finite sets.*$\}$, we obtain its accompanying pure lasso FKS M_L.

Using Algorithms 6 and 8 within M_L, the individual problems of validity, satisfiability, and model checking, each under Boolean reachability semantics, were solved for the three QFLTL formulas φ_1, φ_2, and φ_3. The computational results of these problems are summarized in the following Table 7.

Table 7. Results of solving QFLTL quantitative reachability semantics-related problems within T_M_L.

$\boldsymbol{\varphi_i}$	**Algorithm 6 Validity** $\mathbf{min}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_L)\}$	**Algorithm 6 Satisfiability** $\mathbf{max}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_L)\}$	**Algorithm 8 Model Checking** $\mathbf{arg(min}\{[\![\pi, \varphi_i]\!] \mid \pi \in Path(M_L)\})$
φ_1	0.300	0.770	$\{s_0^\omega\}$
φ_2	0.118	0.088	$\{s_0^\omega\}$
φ_3	0.300	0.685	$\{s_0^\omega\}$

Algorithms 1–8 effectively address the search and decision-making problems of QFLTL. Compared to the model-checking Algorithm in reference [13–17], Algorithms 3, 7, and 8

exhibit improved model-checking results by not only offering the maximum and minimum satisfaction values of the formula but also presenting corresponding counterexample paths, which serve as a foundation for further refinement and enhancement of the model.

These examples fully demonstrate the significant practical application value of both the Boolean reachability semantics and the quantitative reachability semantics of QFLTL, which are more expressive than PoLTL. They can avoid information loss, ensure information synchronization, characterize different weight preferences between path reachability and the satisfaction values of property formulas, and distinguish different weight preferences among property subformulas. Furthermore, the search and decision algorithms for QFLTL presented in the article are effective. Model checking can return counterexample paths, which can serve as an important basis for model refinement and improvement. This is of greater practical significance compared to the PoLTL model-checking algorithms found in the existing work.

7. Conclusions and Future Work

The main contributions of this article are:

(1) It introduces quality constraint operators into fuzzy linear temporal logic (FLTL) and accurately integrates path-reachability information into the path fragments obtained based on formula satisfaction values, considering preference degrees. Furthermore, it embeds the quality constraint operator into each individual property sub-formula to characterize the differing preferences of each sub-attribute towards the overall system attribute requirements;

(2) It provides the syntax of fuzzy linear temporal logic with quality constraints (QFLTL). It explores the practicality of QFLTL, studying its properties and performing a Boolean transformation on it;

(3) It offers an automaton construction algorithm for the Boolean reachable semantics formula of QFLTL and discusses the complexity of the algorithm. Based on this automaton method, we address search and decision problems, such as validity, satisfiability, and model checking under the Boolean reachability semantics of QFLTL. The algorithms for solving these problems are presented, and their complexity is analyzed;

(4) It achieves a further weighted fusion of path reachability and property formula satisfaction values, resulting in a quantitatively reachable QFLTL. Subsequently, it restricts the intelligent system to contain only lasso paths. It describes the validity, satisfiability, model checking, and other search and decision problems of quantitative path-reachability semantics for QFLTL on this specific system, along with the corresponding solving algorithm. The complexity of the algorithm is analyzed;

(5) Finally, it demonstrates the practicality of QFLTL through a medical process example. Concurrently, it verifies that the algorithm proposed in this article for solving the search and decision-making problems of QFLTL is effective.

There is a significant amount of work that needs to be undertaken in this area, including:

(1) developing heuristic searching algorithms specifically for lasso FKSs (finite Kripke structures) in order to diminish the complexity associated with searching and decision-making algorithms that address the quantitative reachability semantics of QFLTL (fuzzy linear temporal logic with quality constraints);

(2) investigating algorithms that can solve searching and decision problems within QFLTL for systems that encompass infinite repeated paths, thereby broadening the applicability of QFLTL in the realm of searching and decision-making;

(3) integrating the concepts of fuzzy time constraints, path-reachability information, and formula satisfaction values, alongside preferences and synchronization and conducting research on FLTL (fuzzy linear temporal logic) with dual constraints of fuzzy time and quality, aiming to enhance the completeness and accuracy of information expression;

(4) addressing the issue of logical decisions within QFLTL, based on the possibilistic decision process [19–21], remains an area that requires further exploration.

Author Contributions: X.Y. conceived the syntax and semantics of QFLTL, delineated its validity, satisfiability, and model checking problems, along with providing pertinent solution algorithms. Y.L. focused on examining the logical attributes of QFLTL and developing automata construction algorithms tailored for QFLTL. S.G. contributed practical instances illustrating QFLTL applications and meticulously proofread the article to ensure adherence to the required writing standards. All authors have read and agreed to the published version of the manuscript.

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), National Key R&Dplan (Grant No: 2020YFC1523305), and Key R&Dand transformation plan of Qinghai Province (Grant No: 2022-QY-203), Scientific and Technological Research Fund of Shangluo University (Grant NO. 20SKY021).

Data Availability Statement: Data are contained within the article.

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

References

1. Bianco, A.; De Alfaro, L. Model checking of probabilistic and nondeterministic systems. In Proceedings of the International Conference on Foundations of Software Technology and Theoretical Computer Science, Bangalore, India, 18–20 December 1995; Springer: Berlin/Heidelberg, Germany, 1995; pp. 499–513.
2. Valiev, M.K.; Dekhtyar, M.I. Complexity of verification of nondeterministic probabilistic multiagent systems. *Autom. Control Comput. Sci.* **2011**, *45*, 390–396. [CrossRef]
3. Baier, C.; Kwiatkowska, M. Model checking for a probabilistic branching time logic with fairness. *Distrib. Comput.* **1998**, *11*, 125–155. [CrossRef]
4. Hart, S.; Sharir, M.; Pnueli, A. Termination of probabilistic concurrent program. *ACM Trans. Program. Lang. Syst.* **1983**, *5*, 356–380. [CrossRef]
5. Kwiatkowska, M.; Norman, G.; Parker, D. PRISM: Probabilistic symbolic model checker. *Comput. Perform. Eval. Tools* **2002**, *24*, 200–204.
6. Klein, J.; Baier, C.; Chrszon, P.; Daum, M.; Dubslaff, C.; Klüppelholz, S.; Märcker, S.; Müller, D. Advances in symbolic probabilistic model checking with PRISM. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Eindhoven, The Netherlands, 9 April 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 349–366.
7. Li, Y.M.; Li, L.J. Model checking of linear-time properties based on possibility measure. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 842–854. [CrossRef]
8. Li, Y.M. Quantitative model checking of linear-time properties based on generalized possibility measures. *Fuzzy Sets Syst.* **2017**, *320*, 17–39. [CrossRef]
9. Zhao, L.; Wu, J.Z. Multi valued model test based on Wu method. *Syst. Sci. Math.* **2008**, *28*, 1020–1029.
10. Chechik, M.; Devereux, B.; Easterbrook, S.; Gurfinkel, A. Multi-valued symbolic model-checking. *ACM Trans. Softw. Eng. Methodol.* **2003**, *12*, 371–408. [CrossRef]
11. Chechik, M.; Devereux, B.; Easterbrook, S. Implementing a multi-valued symbolic model Checker. *Proc. Tools Algorithms Constr. Anal. Syst.* **2001**, *2031*, 92–95.
12. Wang, G.J.; Shi, H.X. Lattice modal proposition logic and its completeness. *Sci. Sin. Inf.-Tionis* **2011**, *41*, 66–76.
13. Frigeri, A.; Pasquale, L.; Spoletini, P. Fuzzy time in linear temporal logic. *ACM Trans. Comput. Log.* **2014**, *15*, 30–53. [CrossRef]
14. Almagor, S.; Boker, U.; Kupferman, O. Formally reasoning about quality. *J. ACM* **2016**, *63*, 1–56. [CrossRef]
15. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [CrossRef]
16. Emerson, E.A. Automata, tableaux, and temporal logics. In Proceedings of the Logics of Programs, Brooklyn, NY, USA, 17–19 June 1985; Springer: Berlin/Heidelberg, Germany, 1985; pp. 79–88.
17. Baier, C.; Katoen, J.P. *Principles of model checking*; MIT Press: Cambridge, MA, USA, 2008.
18. Vardi, M.Y. An automata-theoretic approach to linear temporal logic. *Log. Concurr. Struct. Versus Autom.* **2005**, *1043*, 238–266.
19. Ma, Z.; Gao, Y.; Li, Z.; Li, X.; Liu, Z. Quantitative reachability analysis of generalized possibilistic decision processes. *J. Intell. Fuzzy Syst.* **2023**, *44*, 8357–8373. [CrossRef]
20. 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. [CrossRef]
21. Liu, W.; Wang, J.; He, Q.; Li, Y. Model Checking Computation Tree Logic over Multi-valued Decision Processes and Its Reduction Techniques. *Chin. J. Electron.* **2024**, *34*, 1–13.