Received 9 October 2023, accepted 8 November 2023, date of publication 13 November 2023, date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332367

RESEARCH ARTICLE

Exploring Equivalences in Multi-Valued Systems: Concepts, Properties, and Generation Methods

HUI DENG
School of Statistics, Xi'an University of Finance and Economics, Xi'an 710100, China
e-mail: denghui@xaufe.edu.cn

ABSTRACT This paper investigates equivalences within multi-valued systems. We propose a comprehensive set of concepts related to multi-valued equivalences, such as multi-valued simulation equivalence and multi-valued bisimulation equivalence. We conduct an in-depth examination of the associated theorems and lemmas and provide a detailed comparison of these equivalence techniques to highlight their similarities and differences. To address the issue of state space explosion in multi-valued systems, we introduce the concept of multi-valued quotients based on multi-valued equivalences. Additionally, we present a series of related algorithms to implement and apply these multi-valued quotients.

INDEX TERMS Quasi-Boolean algebra, multi-valued systems, equivalences.

I. INTRODUCTION

Model checking [1], [2], [3] has evolved into an essential technique in formal verification [4], enabling automated analysis of system designs to ensure their compliance with desired properties. However, within model checking, two formidable challenges continue to endure. The first concerns the precise modeling of the uncertainty system [5], while the second focuses on the intricate problem of state space explosion [6], [7].

Confronting the challenge of the precise modeling of the uncertainty system, some scholars have studied the uncertainty model checking methods. Combining the Markov chain, Hart, Sharir, and Baier gave a probabilistic model checking method [8], [9]. However, the probabilistic model checking method does not suit some uncertainty systems that break probabilistic computing characteristics such as non-contradiction law $a \wedge \neg a = 0$ and excluded middle law $a \vee \neg a = 1$. As a result, the theory of model checking based on fuzziness, possibility, and multi-value emerged. Fan et al. explored fuzzy linear time logic [10], and Pan et al. analyzed fuzzy computational tree logic [11]. In addition, Fan et al. also proved the generalization of nondeterministic fuzzy Kripke structures [12]. Li et al. introduced a possibilistic Kripke structure rooted in the possibility measure and researched linear time logic [13], computation tree logic [14], and quantitative model checking methods based on the possibility measure [15]. Chechik made noteworthy contributions to the multi-valued model checking by seamlessly merging theoretical insights with practical applications [16], [17], [18].

In the face of the formidable challenge posed by state space explosion, equivalences are effective, especially in some uncertain systems [19], [20]. For probabilistic systems, Huynh and Tian addressed this challenge by investigating diverse equivalences within probabilistic labeled transition systems [21], and Baier and Katoen introduced probabilistic bisimulation as an equivalence [3]. Focused on the transition relation, Pan et al. introduced lattice-valued bisimulation as an equivalence of lattice-valued transition systems [22]. Integrating the cost problem within possibilistic Kripke structures, Deng et al. introduced the concept of possibilistic cost bisimulation as an equivalence [23].

However, the equivalences of the multi-valued systems are still controversial. Recently, some scholars have investigated the complete residuated lattices [22]. Some well-known algebraic structures, such as Heyting algebras [24], Boolean algebras [25], and Lukasiewicz algebras [26], are all specific cases of complete residuated lattices [27], so complete residuated lattices are universal algebraic structures. This perspective has led many researchers to believe that lattice-valued equivalences cover all equivalences because of the universality of complete residuated lattices. Nonetheless, multi-valued equivalences should be an exception.

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Loconsole.

Specifically, a finite quasi-Boolean algebra [28] with operators $\rightarrow$ defined as $x \rightarrow y = \neg x \vee y$ is not a complete residuated lattice, but the finite quasi-Boolean algebra is the truth value of multi-valued systems. Consequently, current equivalences do not apply to the state explosion in multi-valued systems. Hence, the exploration of equivalences in multi-valued systems is the primary effort of this paper.

This paper is structured as follows: The first section introduces the research background of this paper. The second section revisits some fundamental concepts to establish a foundation for our subsequent analysis. The third section introduces two multi-valued equivalences and clarifies their similarities and distinctions by a comparison. Additionally, it delves into the properties of multi-valued equivalences through related theorems and lemmas and then proposes a novel concept known as multi-valued equivalence composition. The fourth section derives the multi-valued quotients as the minimal form of multi-valued equivalences discussed earlier. Moreover, it details algorithms for generating these multi-valued quotients. As a discussion, the fifth section analyzes the expressions and application of multi-valued equivalences in modeling, offering a comparison with regular ones. Furthermore, it provides the time complexities of multi-valued quotient algorithms across various scenarios. Finally, the concluding section summarizes the innovations and contributions, explores the existing challenges, and proposes the problems for future research.

II. PRELIMINARIES

This subsection offers an overview of the concepts of quasi-Boolean algebra and multi-valued Kripke structures. Please see references [16], [17], [18], [29], [30] for further information.

Definition 1: A quasi-Boolean algebra is a tuple $\mathcal{L} = (L, \sqsubseteq, \sqcup, \sqcap, \neg, 0, 1)$ where$(L, \sqcup, \sqcap)$ is a distributive lattice, 0 and 1 are the bottom element and the top element, $\neg$ is negation operation on L, for any $a, b \in L$ following laws holds:

(1) law of de Morgan: $\neg(a \sqcup b) = \neg a \sqcap \neg b$, $\neg(a \sqcap b) = \neg a \sqcup \neg b$.
(2) law of involution: $\neg\neg a = a$.
(3) law of anti-monotonic: $(a \sqsubseteq b) \Leftrightarrow \neg a \sqsupseteq \neg b$.

A quasi-Boolean algebra transforms into Boolean algebra by two additional laws that apply to each element x within the set S. Firstly, the law of non-contradiction asserts that $a \sqcap \neg a = 0$, ensuring that an element cannot simultaneously possess true and false. Secondly, the law of excluded middle, expressed as $a \sqcup \neg a = 1$, guarantees that for any element, it must either affirm a proposition or its negation, leaving no room for an intermediate truth value.

Definition 2: Given a quasi-Boolean algebra $\mathcal{L}$ and a set S, a multi-valued set on S is defined as a multi-valued function $\mathbb{S} : S \rightarrow \mathcal{L}$, the value of $\mathbb{S}(s)$ denotes the membership of s in S, represented as $s \in S$.

The multi-valued function satisfies the following operations:

(1) Multi-valued intersection: $(\mathbb{S}_1 \cap \mathbb{S}_2)(s) \Leftrightarrow \mathbb{S}_1(s) \sqcap \mathbb{S}_2(s)$.
(2) Multi-valued union: $(\mathbb{S}_1 \cup \mathbb{S}_2)(s) \Leftrightarrow \mathbb{S}_1(s) \sqcup \mathbb{S}_2(s)$.
(3) Multi-valued set inclusion: $\mathbb{S}_1 \subseteq \mathbb{S}_2 \Leftrightarrow \forall s, (\mathbb{S}_1(s) \sqsubseteq \mathbb{S}_2(s))$.
(4) Multi-valued equality: $\mathbb{S}_1 = \mathbb{S}_2 \Leftrightarrow \forall s, (\mathbb{S}_1(s) = \mathbb{S}_2(s))$.

In these operations, $\sqcap$ represents the meet operation, $\sqcup$ represents the join operation, and $\sqsubseteq$ represents the partial order relation in the quasi-Boolean algebra $\mathcal{L}$.

Definition 3: A multi-valued relation R on sets S and T over a quasi-Boolean algebra $\mathcal{L}$ is a function $R : S \times T \rightarrow \mathcal{L}$. R assigns a value in $\mathcal{L}$ to each pair (s, t), where $s \in S$ and $t \in T$. The value $R(s, t)$ represents the relationship between s and t according to the multi-valued relation R.

The multi-valued Kripke structure serves as an essential model of multi-valued systems that expands the formal representation of the traditional Kripke structure. It allows multiple truth values instead of being restricted to binary truth values.

Definition 4: A multi-valued Kripke structure $X = (S, I, R, L, AP, \mathcal{L})$ is a six-tuple, where:

(1) $\mathcal{L} = (L, \sqsubseteq, \sqcup, \sqcap, \neg, 0, 1)$ is a quasi-Boolean algebra that serves as the underlying structure for all multi-valued sets in the model.
(2) AP is a finite set of atomic propositions.
(3) S is a finite set of states.
(4) $I \subseteq S$ represents the set of initial states.
(5) R is the multi-valued transition relation, which maps $(s, t) \in S \times S$ to e, $e \in \mathcal{L}$.
(6) L is the label function, which maps $(s, a) \in S \times AP$ to e, $a \in AP$ and $e \in \mathcal{L}$.

A multi-valued Kripke structure X, let $s_i \in S$ where $0 \leq i \leq n$. $Post(s) = \{\acute{s} \in S | R(s, \acute{s}) = e\}$ denotes the set of successors of state s in X that satisfy the membership e. $Pre(s) = \{\acute{s} \in S | R(\acute{s}, s) = e\}$ denotes the set of predecessors of state s in X that satisfy the membership e. A finite path from state s_0 to s_n is denoted as $\hat{\rho} = s_0 s_1 s_2 \ldots s_n$. An infinite path from state s_0 is denoted as $\rho = s_0 s_1 s_2 \ldots \in s^{\infty}$. $Paths_{fin}(s)$ represents the set of all finite paths originating from state s in X. $Paths(s)$ represents the set of all infinite paths originating from state s in X. $Paths_{fin}(X)$ is the set of all finite paths in X, regardless of the starting state. $Paths(X)$ is the set of all infinite paths in X, regardless of the starting state. The finite trace of a finite path $\hat{\rho}$ can be represented as $trace(\hat{\rho}) = L(s_0, a) \sqcap R(s_0, s_1) \sqcap L(s_1, a) \sqcap R(s_1, s_2) \sqcap \ldots \sqcap R(s_{n-1}, s_n) \sqcap L(s_n, a)$. Similarly, the infinite trace of an infinite path ρ is given by $trace(\rho) = L(s_0, a) \sqcap R(s_0, s_1) \sqcap L(s_1, a) \sqcap R(s_1, s_2) \sqcap \ldots$.

III. EQUIVALENCES IN MULTI-VALUED SYSTEMS

Multi-valued systems are more susceptible to state explosion due to their inherent multi-valued uncertainty. The common approaches for resolving state explosion are abstraction and equivalence. Abstraction simplifies the system by reducing its inherent complexity, which may necessitate omitting specific details, but equivalence is not. It is essential that

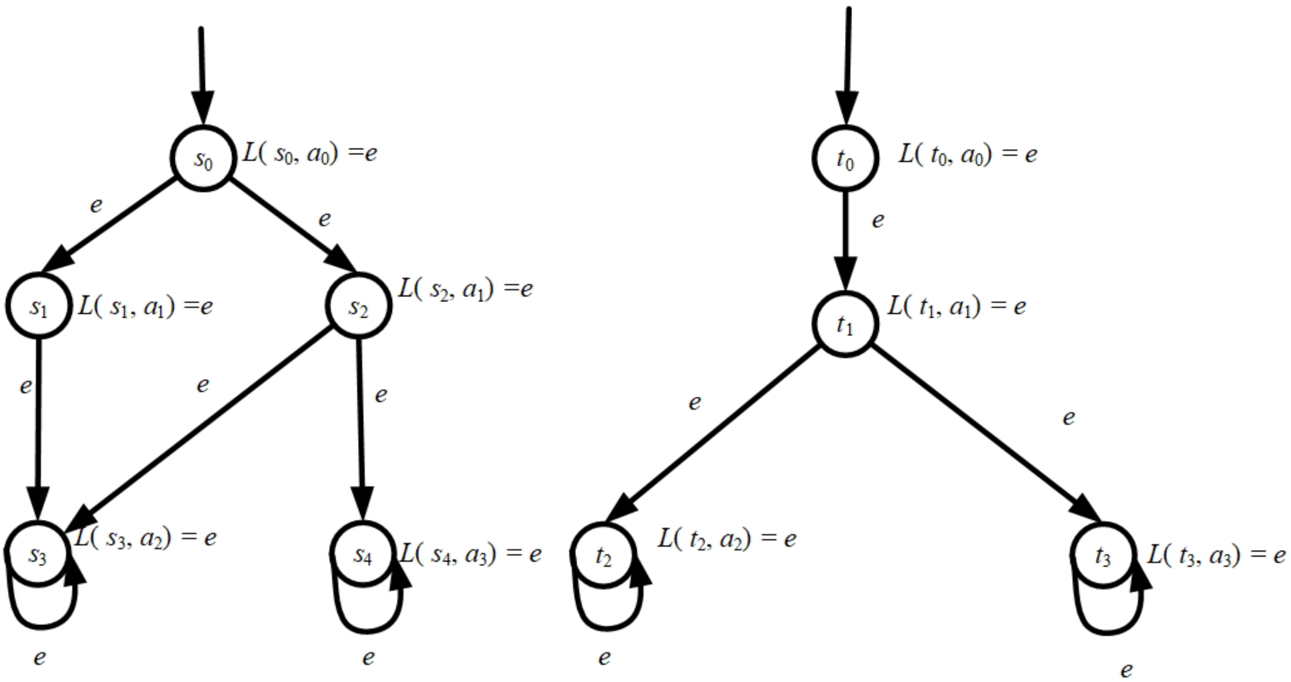

FIGURE 1. Comparison of two multi-valued equivalences.

modeling with a smaller model retains as many critical details as possible for multi-valued systems. To balance simplification and retention, we extend equivalence to multi-valued systems.

A. MULTI-VALUED SIMULATION EQUIVALENCE

Multi-valued simulation equivalence is the strengthening of multi-valued simulation.

Definition 5: Given two multi-valued Kripke structures $X_i = (S_i, I_i, R_i, L_i, AP_i, \mathcal{L})$, where $i = 1, 2$ and $AP_2 \subseteq AP_1$, A multi-valued simulation between X_1 and X_2 is a binary relation "$\preceq$" must meet the following conditions:

(1) $L(s_1, a) = L(s_2, a) = e$ for all $a \in AP$ and $e \in \mathcal{L}$.
(2) If $s_1' \in Post(s_1)$, then there exists $s_2' \in Post(s_2)$ with $s_1' \preceq s_2'$.

If $X_1 \preceq X_2$ and $X_2 \preceq X_1$, X_1 and X_2 are multi-valued simulation equivalence, denoted $X_1 \simeq X_2$.

B. MULTI-VALUED BISIMULATION EQUIVALENCE

Multi-valued bisimulation equivalence is a refinement of multi-valued simulation equivalence. It requires that the satisfaction of each condition must be reciprocal.

Definition 6: Given two multi-valued Kripke structures $X_i = (S_i, I_i, R_i, L_i, AP_i, \mathcal{L})$, where $i = 1, 2$ and $AP_2 \subseteq AP_1$, A multi-valued bisimulation between X_1 and X_2 is a binary relation "$\sim$" must meet the following conditions:

(1) $L(s_1, a) = L(s_2, a) = e, a \in AP, e \in \mathcal{L}$.
(2) If $s_1' \in Post(s_1)$, then there exists $s_2' \in Post(s_2)$ with $s_1' \sim s_2'$.
(3) If $s_2' \in Post(s_2)$, then there exists $s_1' \in Post(s_1)$ with $s_1' \sim s_2'$.

If there exists a relation "$\sim$" from X_1 to X_2, then X_1 and X_2 are multi-valued bisimulation equivalence, denoted as $X_1 \sim X_2$.

C. COMPARISON OF MULTI-VALUED EQUIVALENCES

Multi-valued bisimulation equivalence is a more stringent equivalence relation when compared to multi-valued simulation equivalence. In the context of two multi-valued Kripke structures denoted as X_1 and X_2, the establishment of a multi-valued bisimulation equivalence $X_1 \sim X_2$ naturally leads to the conclusion that X_1 can simulate X_2 ($X_1 \preceq X_2$), and conversely, X_2 can simulate X_1 ($X_2 \preceq X_1$). Consequently, this implies multi-valued simulation equivalence between X_1 and X_2 ($X_1 \simeq X_2$). However, it's essential to note that the reverse does not hold true. Simply being multi-valued simulation equivalence ($X_1 \simeq X_2$) does not guarantee multi-valued bisimulation equivalence ($X_1 \sim X_2$). To illustrate this distinction, we provide an example that showcases a scenario where two structures exhibit multi-valued simulation equivalence but do not meet the stricter criteria of multi-valued bisimulation equivalence.

Consider the multi-valued Kripke structures, denoted as X_1 (on the left) and X_2 (on the right), illustrated in FIGURE 1.

In examining these structures, we first define the multi-valued label functions and multi-valued transition relations:

Multi-valued label functions as follows:

$$L(s_0, a_0) = L(t_0, a_0) = e;$$
$$L(s_1, a_1) = L(s_2, a_1) = L(t_1, a_1) = e;$$
$$L(s_3, a_2) = L(t_2, a_2) = e;$$

$$L(s_4, a_2) = L(t_3, a_3) = e.$$

Multi-valued transition relations as follows:

In X_1, $R(s_0, s_1) = e$ and $R(s_0, s_2) = e$, which can be mirrored by $R(t_0, t_1) = e$ in X_2;

In X_1, $R(s_1, s_3) = e$ and $R(s_2, s_3) = e$, which can be mirrored by $R(t_1, t_2) = e$ in X_2;

In X_1, $R(s_2, s_4) = e$, which can be mirrored by $R(t_1, t_3) = e$ in X_2.

Now, let's explore the reverse direction:

In the reverse direction, we note that the multi-valued transition $R(t_2, t_4)$ in X_2 cannot replicated by any multi-valued transition in X_1. Consequently, X_1 and X_2 fail to satisfy the conditions necessary for multi-valued bisimulation equivalence. However, since X_2 is a subgraph of X_1 and meets the criteria for multi-valued simulation, we have $X_2 \preceq X_1$. Moreover, when considering individual states, we observe that $s_1 \preceq t_1$, $s_2 \preceq t_2$, $s_3 \preceq t_2$, $s_4 \preceq t_3$, and $s_5 \preceq t_4$ which implies $X_1 \preceq X_2$, we can conclude that $X_1 \simeq X_2$, signifying that X_1 and X_2 are multi-valued simulation equivalence.

This comparison and analysis demonstrate that multi-valued bisimulation equivalence is more stringent and meticulous than multi-valued simulation equivalence.

D. PROPERTIES OF MULTI-VALUED EQUIVALENCES

Based on the definitions and discussions provided earlier, we can delve into some properties and characteristics of multi-valued equivalences.

Theorem 7: Reflexivity, Symmetry and Transitivity of multi-valued equivalences.

Proof: Reflexivity, Symmetry, and Transitivity are fundamental properties of multi-valued equivalences. Let's discuss each of these properties in the context of multi-valued equivalences:

Reflexivity: In the context of multi-valued equivalences, reflexivity means that every multi-valued Kripke structure is equivalent to itself. For any multi-valued Kripke structure X, we have $X \sim X$.

Symmetry: Symmetry in multi-valued equivalences implies that if one multi-valued Kripke structure is equivalent to another, the reverse is also true. Formally, if $X_1 \sim X_2$, then it follows that $X_2 \sim X_1$.

Transitivity: Transitivity means that if one multi-valued Kripke structure is equivalent to a second, and the second is equivalent to a third, then the first is also equivalent to the third. Formally, if $X_1 \sim X_2$ and $X_2 \sim X_3$, then it implies $X_1 \sim X_3$.

In summary, multi-valued equivalences satisfy reflexivity, symmetry, and transitivity.

Lemma 8: If X_1 and X_2 satisfied multi-valued equivalences, and $s_{0,1}$ and $s_{0,2}$ represent the multi-valued equivalence states in X_1 and X_2, we can demonstrate that for each finite or infinite path $\rho_1 = s_{0,1}s_{1,1}s_{2,1}\ldots \in Paths(s_{0,1})$ within the set of paths originating from state $s_{0,1}$ in X_1, there exists a corresponding path $\rho_2 = s_{0,2}s_{1,2}s_{2,2}\ldots \in Paths(s_{0,2})$ within the set of paths originating from state $s_{0,2}$ in X_2 with the same length, such that $s_{i,1}$ and $s_{i,2}$ are the multi-valued equivalence states, and this holds for all $i \geq 0$.

Proof: We can prove this statement using an inductive argument:

Base Case: For $i = 0$, both paths ρ_1 and ρ_2 start at their respective multi-valued equivalence states, so $s_{0,1}$ and $s_{0,2}$ are the multi-valued equivalence states for both paths. Given that X_1 and X_2 are multi-valued equivalence structures, $s_{0,1}$ and $s_{0,2}$ are indeed the multi-valued equivalence states.

Inductive Step: Assume that for some $k \geq 0$, the statement holds for all i up to k, i.e., $s_{i,1}$ and $s_{i,2}$ are the multi-valued equivalence states for all i from 0 to k.

For $i = k + 1$, consider ρ_1 and ρ_2 as follows:

$$\rho_1 = s_{0,1}s_{1,1}s_{2,1}\ldots s_{k,1}s_{k+1,1}.$$
$$\rho_2 = s_{0,2}s_{1,2}s_{2,2}\ldots s_{k,2}s_{k+1,2}.$$

Since we assumed that the statement holds for $i = k$, we know that $s_{k,1}$ and $s_{k,2}$ are the multi-valued equivalence states for both paths.

Now show that $s_{k+1,1}$ and $s_{k+1,2}$ are the multi-valued equivalence states. Since X_1 and X_2 are multi-valued equivalence structures, and $s_{k,1}$ and $s_{k,2}$ are multi-valued equivalence states, we can apply the definition of multi-valued equivalence to conclude that $s_{k+1,1}$ and $s_{k+1,2}$ are also the multi-valued equivalence states.

By induction, we have shown that for any $i \geq 0$, $s_{i,1}$ and $s_{i,2}$ are the multi-valued equivalence states. It demonstrates that for any finite or infinite path starting from $s_{0,1}$ in X_1, there exists a corresponding one starting from $s_{0,2}$ in X_2 with the same length, and the states along these paths are the multi-valued equivalence states.

Theorem 9: If X_1 and X_2 satisfied multi-valued equivalences, it follows that the trace of X_1 is multi-valued equivalent to the trace of X_2, expressed as $trace(X_1) = trace(X_2)$.

Proof: If X_1 and X_2 satisfied multi-valued equivalences, we have established the following:

1. Any path $\rho_1 = s_{0,1}s_{1,1}s_{2,1}\ldots$ in X_1 can deduce a same length path $\rho_2 = s_{0,2}s_{1,2}s_{2,2}\ldots$ in X_2.

2. $L_1(s_{i,1}, a) = L_2(s_{i,2}, a)$, and $R_1(s_{i,1}, s_{i+1,1}) = R_2(s_{i,2}, s_{i+1,2})$ by the conditions of multi-valued equivalence.

Now, let's apply this to the traces of X_1 and X_2, consider the trace $trace(X_1)$ of X_1. It represents the set of all possible traces generated by X_1 during its execution. By the first point, for every path ρ_1 in X_1, there exists a corresponding ρ_2 in X_2 such that $s_{i,1}$ and $s_{i,2}$ are multi-vauled equivalence states, and by the second point, we have $L_1(s_{i,1}, a) = L_2(s_{i,2}, a)$ and $R_1(s_{i,1}, s_{i+1,1}) = R_2(s_{i,2}, s_{i+1,2})$ for all i, it implies that for any path ρ_1 in X_1, there exists a corresponding ρ_2 in X_2 such that the traces are equivalent. By symmetry, for any path ρ_2 in X_2, there exists a corresponding ρ_1 in X_1 such that the traces are equivalent.

In conclusion, if X_1 and X_2 satisfied multi-valued equivalences, then $trace(X_1) = trace(X_2)$.

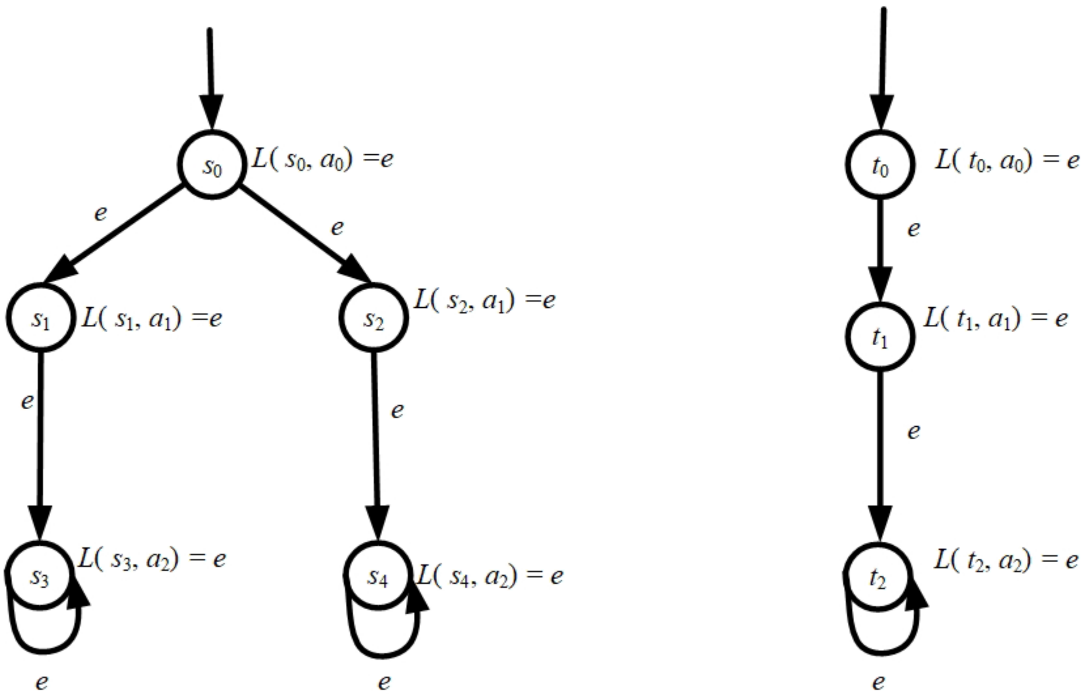

FIGURE 2. Multi-valued simulation quotient.

E. MULTI-VALUED EQUIVALENCE COMPOSITION

Definition 10: For two multi-valued Kripke structures $X_i = (S_i, I_i, R_i, L_i, AP, \mathcal{L}), i = 1, 2$ that satisfy the multi-valued equivalences, the multi-valued equivalence composition $X_1 \oplus X_2 = (S_1 \uplus S_2, I_1 \cup I_2, R_1 \cup R_2, L, AP, \mathcal{L})$.

Here, $\uplus$ represents the disjoint union operation, $L(s, a) = L_i(s, a)$ if $s \in S_i$. In the multi-valued equivalence composition, the state space is the union of the state spaces of X_1 and X_2, the initial states and the transition relation consist of the initial states and the union of the transition relations of both, respectively, and the labeling function is determined by $L(s, a)$ which depends on the corresponding structure.

IV. MULTI-VALUED QUOTIENTS

To simplify complex structures and highlight the essential behavioral characteristics of multi-valued systems, multi-valued Kripke structures can be transformed into a more concise form using multi-valued equivalences, known as multi-valued quotients. The multi-valued quotients divide the state space into several equivalence classes and treat each equivalent class as a single state. Depending on the specific multi-valued equivalences used, multi-valued quotients divide into a multi-valued simulation quotient and a multi-valued bisimulation quotient. This section focuses on defining the multi-valued quotients and developing some algorithms for generating them.

A. MULTI-VALUED SIMULATION QUOTIENT

Definition 11: In a multi-valued Kripke structure $X = (S, I, R, L, AP, \mathcal{L})$ with multi-valued simulation equivalence $\simeq$, the multi-valued simulation quotient is defined as $X/_{\simeq} = (S/_{\simeq}, I_{\simeq}, R_{\simeq}, L_{\simeq}, AP, \mathcal{L})$.

The components of the multi-valued simulation quotient are defined as follows:

(1) $S/_{\simeq} = \{[s]_{\simeq} | s \in S\}$ represents multi-valued simulation quotient space, where $[s]_{\simeq} = \{\acute{s} \in S | s \simeq \acute{s}\}$.
(2) $I_{\simeq} = \{[s]_{\simeq} | I \subseteq S\}$ denotes multi-valued simulation equivalent initial states of I in X.
(3) $R_{\simeq}$ is multi-valued simulation equivalent transition relation defined as $\frac{R(s_1, s_2)=e}{R_{\simeq}([s_1]_{\simeq}, [s_2]_{\simeq})=e}$, where $e \in \mathcal{L}$.
(4) $L_{\simeq}([s]_{\simeq}, a) = L(s, a) = e$ represents multi-valued simulation equivalent labeling function, where $a \in AP$ and $e \in \mathcal{L}$.
(5) AP is a set of atomic propositions.
(6) $\mathcal{L}$ is a quasi-Boolean algebra.

The multi-valued simulation quotient provides a simplified representation of the original multi-valued Kripke structure by multi-valued simulation equivalence, preserving relevant properties while reducing the complexity of the multi-valued system.

Let's explore an illustrative example of a multi-valued simulation quotient. Based on the updated information, the multi-valued Kripke structure X_2 (on the right) can be considered as the multi-valued simulation quotient of the multi-valued Kripke structure X_1 (on the left), as depicted in FIGURE 2.

Consider the multi-valued label functions:

$$L(s_0, a_0) = L(t_0, a_0) = e;$$
$$L(s_1, a_1) = L(s_2, a_1) = L(t_1, a_1) = e;$$

$$L(s_3, a_2) = L(s_4, a_2) = L(t_2, a_2) = e.$$

Now, let's consider the multi-valued transition relations:

$R(t_0, t_1) = e$ in X_2 can be mimicked by $R(s_0, s_1) = e$ and $R(s_0, s_2) = e$ in X_1;

$R(t_1, t_2) = e$ in X_2 can be mimicked by $R(s_1, s_3) = e$ and $R(s_2, s_4) = e$ in X_1.

Since X_2 is a subgraph of X_1 and satisfies the conditions of multi-valued simulation, and since each state in X_1 can mimicked by a state in X_2 and X_2 cannot be further reduced, we can conclude that X_2 is the multi-valued simulation quotient of X_1.

B. MULTI-VALUED SIMULATION QUOTIENT ALGORITHM

The process for implementing the multi-valued simulation quotient is Algorithm 1.

Algorithm 1 Multi-Valued Simulation Quotient Algorithm

Input: $X = (S, I, R, L, AP, \mathcal{L})$
Output: $X/_{\simeq} = (S/_{\simeq}, I_{\simeq}, R_{\simeq}, L_{\simeq}, AP, \mathcal{L})$

1: initialize $E = \{(s_1, s_2) \mid L(s_1)(a) = L(s_2)(a) = e\}$;
2: **while** E is not a multi-valued simulation **do**
3: select $(s_1, s_2) \in E$ such that $R(s_1, \acute{s}_1) = e$, and there is no successor state $\acute{s}_2$ of s_2 satisfying $R(s_2, \acute{s}_2) = e$ and $(\acute{s}_1, \acute{s}_2) \in E$;
4: $E = E \setminus \{(s_1, s_2)\}$;
5: **end while**
6: restructure X into $X/_{\simeq}$ by AP and $\mathcal{L}$;
7: **return** $X/_{\simeq} = (S/_{\simeq}, I_{\simeq}, R_{\simeq}, L_{\simeq}, AP, \mathcal{L})$;

The basic idea of the multi-valued simulation quotient algorithm can be summarized as follows:

(1) Initialization(step 1): Commence by establishing the initial equivalence relation $E = \{(s_1, s_2) \mid L(s_1)(a) = L(s_2)(a) = e\}$, where $a \in AP$ and $e \in \mathcal{L}$.
(2) Iterative(steps 2-5): Continuously execute the subsequent loop as the relation E fails to fulfill the criteria of a multi-valued simulation relation. Choose a pair (s_1, s_2) from the set E such that $R(s_1, \acute{s}_1) = e$, and there exists no successor state $\acute{s}_2$ of s_2 that satisfies $R(s_2, \acute{s}_2) = e$ and $(\acute{s}_1, \acute{s}_2) \in E$, where e represents an element within the provided quasi-Boolean algebra $\mathcal{L}$. Eliminate the selected pair (s_1, s_2) from the relation E.
(3) Restructure(step 6): Provide the resultant relation E as the state space of the multi-valued simulation quotient, and combine AP and $\mathcal{L}$ restructure as a multi-valued Kripke structure.
(4) Outcome(step 7): Return a new multi-valued Kripke structure, as a result of the process, as the multi-valued simulation quotient.

The time complexity of the multi-valued simulation quotient algorithm is $O(|M| \cdot |S|^3 \cdot |N|^2)$, where $|M|$ represents the number of edges in the states graph $G(X)$, $|S|$ represents the number of states in the state space, and $|N|$ represents the number of memberships in $\mathcal{L}$.

C. MULTI-VALUED BISIMULATION QUOTIENT

Definition 12: In a multi-valued Kripke structure $X = (S, I, R, L, AP, \mathcal{L})$ with multi-valued states bisimulation equivalence $\sim$, the multi-valued bisimulation quotient is defined as $X/_{\sim} = (S/_{\sim}, I_{\sim}, R_{\sim}, L_{\sim}, AP, \mathcal{L})$.

The components of the multi-valued simulation quotient are defined as follows:

(1) $S/_{\sim} = \{[s]_{\sim} | s \in S\}$ represents multi-valued bisimulation quotient space, where $[s]_{\sim} = \{\acute{s} \in S | s \sim \acute{s}\}$.
(2) $I_{\sim} = \{[s]_{\sim} | I \subseteq S\}$ denotes multi-valued bisimulation equivalent initial states of I in X.
(3) $R_{\sim}$ is a multi-valued bisimulation equivalent transition relation defined as $\frac{R(s_1, s_2) = e}{R_{\sim}([s_1]_{\sim}, [s_2]_{\sim}) = e}$, where $e \in \mathcal{L}$.
(4) $L_{\sim}([s]_{\sim}, a) = L(s, a) = e$ represents multi-valued bisimulation equivalent labeling function, where $a \in AP$ and $e \in \mathcal{L}$.
(5) AP is a set of atomic propositions.
(6) $\mathcal{L}$ is a quasi-Boolean algebra.

The multi-valued bisimulation quotient also offers a simplified representation of the original structure, preserving relevant properties while reducing the complexity of the multi-valued system. Let's delve into an illustrative example of a multi-valued bisimulation quotient. FIGURE 3 depicts the multi-valued Kripke structure X_1 (on the left) and its multi-valued bisimulation quotient X_2 (on the right).

Let's consider the multi-valued label functions:

$$L(s_0, a_0) = L(t_0, a_0) = e;$$
$$L(s_1, a_1) = L(s_2, a_1) = L(s_3, a_1) = L(t_1, a_1) = e;$$
$$L(s_4, a_2) = L(t_2, a_2) = e;$$
$$L(s_5, a_2) = L(t_3, a_2) = e.$$

Additionally, the multi-valued transition relations are as follows:

In X_1, $R(s_0, s_1) = e$, $R(s_0, s_2) = e$, and $R(s_0, s_3) = e$, which can be mimicked by $R(t_0, t_1) = e$ in X_2.

In X_1, $R(s_2, s_4) = e$, which can be mimicked by $R(t_1, t_2) = e$ in X_2.

In X_1, $R(s_2, s_5) = e$, which can be mimicked by $R(t_1, t_3) = e$ in X_2.

Now, let's consider the reverse direction:

In X_2, $R(t_0, t_1) = e$ can be mimicked by $R(s_0, s_1) = e$, $R(s_0, s_2) = e$, and $R(s_0, s_3) = e$ in X_1.

In X_2, $R(t_1, t_2) = e$ can be mimicked by $R(s_2, s_4) = e$ in X_1.

In X_2, $R(t_1, t_3) = e$ can be mimicked by $R(s_2, s_5) = e$ in X_1.

In formal terms, if X_2 is not simplified any further and $X_1 \sim X_2$, it concluded that X_2 is indeed the multi-valued bisimulation quotient of X_1. This multi-valued bisimulation quotient provides a more concise representation of the original multi-valued Kripke structure while preserving the relevant behavioral properties.

The multi-valued bisimulation equivalence captures more subtle behavioral similarities between states in a multi-valued

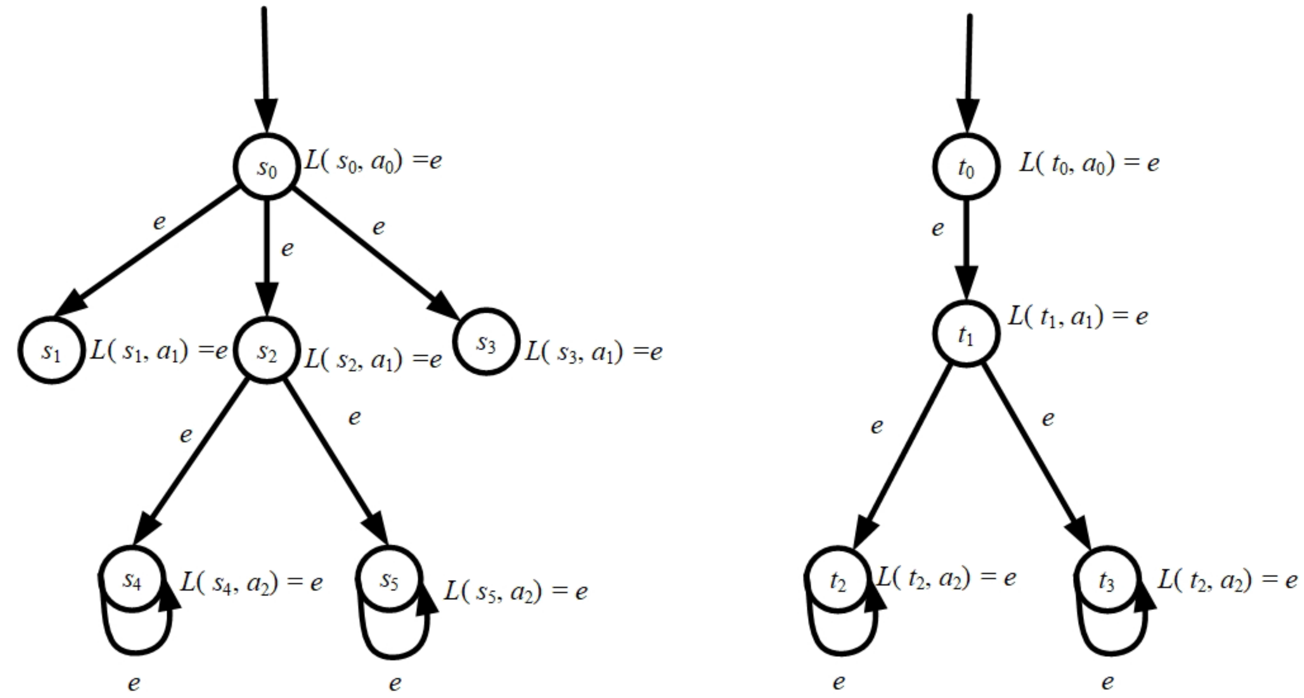

FIGURE 3. Multi-valued bisimulation quotient.

Kripke structure compared to multi-valued simulation equivalence. As a result, the multi-valued bisimulation quotient provides a more condensed representation of the original while preserving the necessary behavioral properties.

D. MULTI-VALUED BISIMULATION QUOTIENT ALGORITHM

The process for implementing the multi-valued bisimulation quotient is Algorithm 2.

The multi-valued bisimulation quotient algorithm consists of multi-valued decision tree generation and multi-valued sets partitions. These two main components bridge the gap between theoretical concepts and their practical computational implementations, ensuring clarity and efficiency. As an initialization step, the multi-valued decision tree generation divides the states into several equivalent classes based on the same multi-valued label function. The multi-valued bisimulation quotient is then obtained by treating these equivalent classes as different multi-valued sets and subsequently refining these multi-valued sets.

In summary, the core idea of the multi-valued bisimulation quotient algorithm is as follows:

(1) Generation(steps 1-16): In the multi-valued decision tree, the various branches represent different label function memberships, and the height represents the number of elements of the atomic propositions set. For example, if $AP = \{a_1, a_2\}$, the height is 2. The vertices at 1st depth in the tree represent $L(s_i, a_1) = e$, and the vertices at 2ed depth represent $L(s_i, a_2) = e$, where $s_i \in S$ and $e \in \mathcal{L}$. Ultimately, the state space is divided into various leaf nodes by different memberships e, and each leaf node represents the equivalent classes w with the same label function.

(2) Initialization(step 18, step 23): Initialize two initial partitions, i.e., initial preorder partitions $\Pi_{pre} := w$ and initial successor partitions $\Pi_{post} := w$, as two starting points for subsequent iterations.

(3) Iterative(steps 19-22, steps 24-27): Iteratively refine the preorder partitions Π_{pre} and successor partitions Π_{post}. Continue these two processes until no further merges or splits are possible, resulting in two partitions ultimately.

(4) Combination(step 28): Combine the ultimate partitions of preorder partitions Π_{pre} and successor partitions Π_{post} into a single refinement partition $\Pi_{refinement}$.

(5) Restructure(step 29): Use $\Pi_{refinement}$, AP, and $\mathcal{L}$ to restructure and generate a new multi-valued Kripke structure.

(6) Outcome(step 30): Provide the newly generated multi-valued Kripke structure as the output, representing the multi-valued bisimulation quotient.

For each state $s \in S$, traversing the multi-valued decision tree from root to leaf takes time $O(|AP| \cdot |N|)$. Therefore, the time complexity of generation is $O(|S| \cdot |AP| \cdot |N|)$, where $|AP|$ represents the number of values in the label function AP, $|S|$ represents the number of states in the state space, and $|N|$ represents the number of memberships in $\mathcal{L}$. According to the given information, for the preorder partitions, the time complexity for each state $s \in C$ is $O(|Pre(s)| \cdot |N|)$, and for

TABLE 1. A comparison between multi-valued equivalences and regular equivalences.

Aspect	Multi-Valued Equivalences	Regular Equivalences
Truth Values	Between "0" and "1"	Limited to "0" and "1"
Label Function Values	Quasi-Boolean algebra	Typically binary
Transition Relation	Quasi-Boolean algebra	Typically binary
Applicability	Multi-Valued Uncertain System Models and Regular System Models	Regular System Models
Expressiveness	More Expressive	Less Expressive

Algorithm 2 Multi-Valued Bisimulation Quotient Algorithm

Input: $X = (S, I, R, L, AP, \mathcal{L})$
Output: $X/_\sim = (S/_\sim, I_\sim, R_\sim, L_\sim, AP, \mathcal{L})$

new(v_0)
for all $s \in S$ **do**
  $v = v_0$
  for all $i = 1, \ldots, n-1$ **do**
    for all $a \in AP$ **do**
      if $L(s, a) = e$ **then**
        new(branch(v))
        v:=branch(v)
      end if
    end for
  end for
  if $L(s, a) = e$ **then**
    new(branch(v))
    states(branch(v)):=states(branch(v))$\cup s$
  end if
end for
w :=states(branch(v))
$\Pi_{pre} := w$
while there exists a multi-valued pioneer splitter for Π_{pre} **do**
  choose a multi-valued pioneer splitter C for Π
  $\Pi_{pre} := \bigcup_{B\in\Pi} (\bigcup_{e\in\mathcal{L}} (B \cap Pre(C)))$
end while
$\Pi_{post} := w$
while there exists a multi-valued successor splitter for Π_{post} **do**
  choose a multi-valued successor splitter C for Π
  $\Pi_{post} := \bigcup_{B\in\Pi} (\bigcup_{e\in\mathcal{L}} (B \cap Post(C))$
end while
$\Pi_{refinement} = \Pi_{pre} \cap \Pi_{post}$;
restructure X into $X/_\sim$ by $\Pi_{refinement}$, AP and $\mathcal{L}$;
return $X/_\sim = (S/_\sim, I_\sim, R_\sim, L_\sim, AP, \mathcal{L})$;

the successor partitions, the time complexity for each state $s \in C$ is $O(|Post(s)| \cdot |N|)$. Therefore, for all states, the time complexity of the preorder partitions is $O(\sum_{s\in S} |Pre(s)| \cdot |N|) = O(|Pre(C)| \cdot |N|)$, and the time complexity of the successor partitions is $O(\sum_{s\in S} |Post(s)| \cdot |N|) = O(|Post(C)| \cdot |N|)$. As a result, the overall time complexity of the partition operators is $O(|Pre(C)| \cdot |N| + |Post(C)| \cdot |N|)$. Thus, the time complexity of the multi-valued bisimulation quotient algorithm is $O(|S| \cdot |AP| \cdot |N| + |Pre(C)| \cdot |N| + |Post(C)| \cdot |N|)$.

TABLE 2. The time complexities of the multi-valued simulation quotient algorithm.

label functions	transition relations	time complexities
$e \neq 1$	$e \neq 1$	$O(\|M\| \cdot \|S\|^3 \cdot \|N\|^2)$
$e \neq 1$	$e = 1$	$O(\|M\| \cdot \|S\|^3 \cdot \|N\|)$
$e = 1$	$e \neq 1$	$O(\|M\| \cdot \|S\|^3 \cdot \|N\|)$
$e = 1$	$e = 1$	$O(\|M\| \cdot \|S\|^3)$

V. DISCUSSION

In this paper, we introduced the multi-valued equivalences based on quasi-Boolean algebra. These multi-valued equivalences extend the traditional binary truth values of "0" or "1" to cover a continuous range between "0" and "1".

(1) Extension of label function values: We expanded the values of label functions to quasi-Boolean algebra, allowing them to take on between "0" and "1". This extension provides a more nuanced and expressive representation of label function values, accommodating a range of possibilities.

(2) Extension of transition relation values: Similarly, the values of transition relations were extended to quasi-Boolean algebra, covering the continuous range between "0" and "1". This extension enables a more flexible and expressive representation of transitions, especially in scenarios involving multi-valued uncertain models.

These extensions make multi-valued equivalences more expressive and applicable in modeling and analyzing systems. For a side-by-side comparison with regular equivalences, please refer to TABLE 1.

As a result, all multi-valued equivalence methods are not limited to multi-valued uncertain models but are also effectively applied to regular model problems. Concepts like multi-valued quotients play a significant role in multi-valued equivalence techniques as they minimize multi-valued uncertain systems for verification efficiency. A minimized multi-valued model replaces the original multi-valued model for verification purposes. When $e = 1$, the multi-valued quotients reduce to the regular quotients. Therefore, the multi-valued quotient algorithms also apply to solving the regular quotients.

For a detailed breakdown of the time complexities associated with multi-valued simulation quotient algorithms in various scenarios, please refer to TABLE 2.

TABLE 3. The time complexities of the multi-valued bisimulation quotient algorithm.

label functions	transition relations	time complexities
$e \neq 1$	$e \neq 1$	$O(\lvert S\rvert \cdot \lvert AP\rvert \cdot \lvert N\rvert + \lvert Pre(C)\rvert \cdot \lvert N\rvert + \lvert Post(C)\rvert \cdot \lvert N\rvert)$
$e \neq 1$	$e = 1$	$O(\lvert S\rvert \cdot \lvert AP\rvert \cdot \lvert N\rvert + \lvert Pre(C)\rvert + \lvert Post(C)\rvert)$
$e = 1$	$e \neq 1$	$O(\lvert S\rvert \cdot \lvert AP\rvert + \lvert Pre(C)\rvert \cdot \lvert N\rvert + \lvert Post(C)\rvert \cdot \lvert N\rvert)$
$e = 1$	$e = 1$	$O(\lvert S\rvert \cdot \lvert AP\rvert + \lvert Pre(C)\rvert + \lvert Post(C)\rvert)$

For a detailed breakdown of the time complexities associated with multi-valued bisimulation quotient algorithms in various scenarios, please refer to TABLE 3.

VI. CONCLUSION

Multi-valued systems are more susceptible to the challenge of state space explosion due to the inherent multi-valued uncertainty. In light of this, multi-valued equivalences are crucial for addressing this issue. This paper has introduced several such equivalences tailored for multi-valued systems, including multi-valued simulation equivalence, multi-valued bisimulation equivalence, multi-valued simulation quotient, and multi-valued bisimulation quotient.

However, it's important to note that some algorithms of multi-valued simulation quotients are initial endeavors, and there is room for further refinement. The number of edges in the state graph, the total count of states in the state space, and the number of memberships in $\mathcal{L}$ significantly impact computational efficiency. In our forthcoming research, we intend to delve deeper into these algorithms and endeavor to develop enhanced versions that can better tackle the challenges posed by multi-valued systems.

REFERENCES

[1] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, *Handbook of Model Checking*. Berlin, Germany: Springer, 2014.

[2] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[3] C. Baier and J. P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[4] R. M. Keller, "Formal verification of parallel programs," *Commun. ACM*, vol. 19, no. 7, pp. 371–384, Jul. 1976.

[5] M. A. Bilal and J. K. George, *Uncertainty Modeling and Analysis in Engineering and the Sciences*. Boca Raton, FL, USA: CRC Press, 2006.

[6] J. F. Groote, T. W. D. M. Kouters, and A. Osaiweran, "Specification guidelines to avoid the state space explosion problem," *Softw. Test., Verification Rel.*, vol. 25, no. 1, pp. 4–33, Jan. 2015.

[7] K. McMillan, *Symbolic Model Checking: An Approach to the State Explosion Problem*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 1992.

[8] S. Hart and M. Sharir, "Probabilistic propositional temporal logics," *Inf. Control*, vol. 70, nos. 2–3, pp. 97–155, Aug. 1986.

[9] C. Baier and M. Kwiatkowska, "Model checking for a probabilistic branching time logic with fairness," *Distrib. Comput.*, vol. 11, no. 3, pp. 125–155, Aug. 1998.

[10] Y. H. Fan and Y. M. Li, "The realizability of fuzzy linear temporal logic," *Acta Electronica Sinica*, vol. 46, no. 2, pp. 341–346, 2018.

[11] H. Pan, Y. Li, Y. Cao, and Z. Ma, "Model checking fuzzy computation tree logic," *Fuzzy Sets Syst.*, vol. 262, pp. 60–77, Mar. 2015.

[12] Y. H. Fan, Y. M. Li, and H. Y. Pan, "Computation tree logic model checking for nondeterminisitc fuzzy Kripke structure," *Acta Electronica Sinica*, vol. 46, no. 1, pp. 152–159, 2018.

[13] Y. Li and L. Li, "Model checking of linear-time properties based on possibility measure," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 842–854, Oct. 2013.

[14] Y. Li, Y. Li, and Z. Ma, "Computation tree logic model checking based on possibility measures," *Fuzzy Sets Syst.*, vol. 262, pp. 44–59, Mar. 2015.

[15] Y. Li and Z. Ma, "Quantitative computation tree logic model checking based on generalized possibility measures," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2034–2047, Dec. 2015.

[16] M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel, "Multi-valued symbolic model checking," *ACM Trans. Softw. Eng. Methodol.*, vol. 12, no. 4, pp. 371–408, 2003.

[17] M. Chechik, A. Gurfinkel, B. Devereux, A. Lai, and S. Easterbrook, "Data structures for symbolic multi-valued model-checking," *Formal Methods Syst. Design*, vol. 29, no. 3, pp. 295–344, Oct. 2006.

[18] Y. Li, L. Lei, and S. Li, "Computation tree logic model checking based on multi-valued possibility measures," *Inf. Sci.*, vol. 485, pp. 87–113, Jun. 2019.

[19] M. Lange, E. Lozes, and M. V. Guzmán, "Model-checking process equivalences," *Theor. Comput. Sci.*, vol. 560, pp. 326–347, Dec. 2014.

[20] P. Kanellakis and S. Smolka, "CCS expressions, finite state processes, and three problems of equivalence," *Inf. Comput.*, vol. 86, no. 1, pp. 43–68, 1983.

[21] T. Huynh and L. Tian, "On some equivalence relations for probabilistic processes," *Fundam. Inform.*, vol. 17, no. 3, pp. 211–234, 1992.

[22] H. Pan, M. Zhang, and Y. Chen, "Bisimulation for lattice-valued transition systems," in *Proc. 6th Int. Symp. Theor. Aspects Softw. Eng.* Beijing, China, Jul. 2012, pp. 279–282.

[23] H. Deng, Y. Zhang, and Z. Huang, "Possibilistic cost computation tree logic and related equivalence, abstraction technique," *IEEE Access*, vol. 11, pp. 2478–2487, 2023.

[24] R. Belohlavek, *Fuzzy Relational Systems: Foundations and Principles*. New York, NY, USA: Springer, 2012.

[25] B. Jonnson and A. Tarski, "Boolean algebras with operators," *Amer. J. Math.*, vol. 74, no. 1, p. 127, Jan. 1952.

[26] T. Kroupa, "States in Łukasiewicz logic correspond to probabilities of rational polyhedra," *Int. J. Approx. Reasoning*, vol. 53, no. 4, pp. 435–446, Jun. 2012.

[27] D. Qiu, "Automata theory based on complete residuated lattice-valued logic," *Sci. China F, Inf. Sci.*, vol. 44, no. 6, pp. 419–429, Dec. 2001.

[28] J. M. Dunn, "A relational representation of quasi-Boolean algebras," *Notre Dame J. Formal Log.*, vol. 23, no. 4, Oct. 1982.

[29] M. Chechik, S. Easterbrook, and V. Petrovykh, "Model-checking over multi-valued logics," in *Formal Methods for Increasing Software Productivity* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2021, pp. 72–98.

[30] L. Bloc and P. Borowik, *Multi-Valued Logics*. Berlin, Germany: Springer-Verlag, 1992.

HUI DENG received the B.Sc. degree in computer science and technology from Xi'an Polytechnic University, Xi'an, China, in 2007, the M.Sc. degree in computer software and theory from Shaanxi Normal University, Xi'an, in 2013, and the Ph.D. degree in control theory and control engineering from Xidian University, Xi'an, in 2020.

From 2007 to 2010, he was an Assistant Editor with the Fourth Military Medical University Press, Xi'an. From 2012 to 2017, he was a Software Engineer with IBM China Development Laboratory, Xi'an. He is currently a Lecturer with the Department of Data Science and Big Data Technology, School of Statistics, Xi'an University of Finance and Economics, Xi'an. His research interests include big data analysis, model checking, artificial intelligence, and computational intelligence.