International Journal of Innovative Computing, Information and Control
Volume **12**, Number **5**, October **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1629–1643

MULTI-ARY α-SEMANTIC RESOLUTION AUTOMATED REASONING BASED ON A LATTICE-VALUED PROPOSITION LOGIC LP(X)

Yi Liu[1,2], Hairui Jia[1] and Yang Xu[1]

[1]Sichuan Provincial System Reliability Automatic Verification Laboratory
Southwest Jiaotong University
No. 111, North Section 1, 2nd Ring Road, Chengdu 610031, P. R. China
xuyang@home.swjtu.edu.cn

[2]Data Recovery Key Lab of Sichuan Province
Neijiang Normal University
No. 705, Dongtong Road, Neijiang 641000, P. R. China
liuyiyl@126.com

Received March 2016; revised July 2016

Abstract. *This paper focuses on multi-ary α-semantic resolution automated reasoning method based on multi-ary α-resolution principle for lattice-valued propositional logic LP(X) with truth-value in lattice implication algebras. The definitions of the multi-ary α-semantic resolution and multi-ary α-semantic resolution deduction in lattice-valued propositional logic LP(X) are given, respectively, and the soundness and completeness are gotten. An algorithm of multi-ary α-semantic resolution method is constructed; the soundness and completeness of multi-ary α-semantic resolution algorithm are also obtained. This work will provide a theoretical foundation for the more efficient resolution based automated reasoning in lattice-valued logic.*

Keywords: Lattice implication algebra, Lattice-valued propositional logic, Automated reasoning, Multi-ary α-semantic resolution

1. **Introduction.** Resolution principle was introduced by Robinson [4] in 1965, and it revolutionized the field of automated reasonings as mechanizable method for detecting the unsatisfiability of a given set of formulae in classical first-order logic. Since then, many refinements of resolution methods have been proposed by researchers to cut down the search space and increase efficiency. Semantic resolution [5], introduced by Slagle in 1967, is one of the most important refinements of resolution principle in classical logic. Semantic resolution method can improve the efficiency of reasoning by reducing the redundant clauses with restraining the type of clauses and the order of literals participated in resolution procedure. Subsequently, many scholars give various kinds of improved semantic resolution methods [1], which can effectively improve the efficiency of automated reasoning.

Non-classical logics have been widely used in computer science, AI and logic programming. Automated theorem proving (or automated reasoning) based on non-classical logic is also an active field of non-classic logic. Lattice-valued logic with truth-value in a lattice implication algebra, an important non-classical logic, is also widely investigated due to the fact that it can process effectively the incomparability. There have also been investigations of resolution-based automated reasoning in lattice-valued logic with truth-value in lattice implication algebras (LIAs) (e.g., among others, [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]). Correspondingly, the resolution methods based on lattice-valued logic have some new features, for example, (a) resolution is proceeded at a different truth-valued level α chosen from the truth-valued field – LIA, in each resolution of α-resolution deduction, choosing 2 generalized literals, which contain constants and implicative connectives, to take part in the resolution; so, the α-resolution is also called 2-ary α-resolution; (b) Comparing with the resolution based on classical logic, owing to the fact that the structure of generalized literal in lattice-valued logic is very complex, it is not easy to directly judge if two generalized literals are α-resolvent. Therefore, a 2-ary α-resolution principle for a lattice-valued propositional logic LP(X) has been proposed in [7], which can be used to prove whether a lattice-valued logical formula in LP(X) is false at a truth-value level α (i.e., α-false) or not, and the theorems of soundness and completeness for the 2-ary α-resolution principle were also proved. In addition, [8] extends the 2-ary α-resolution principle for LP(X) to the corresponding lattice-valued first-order logic LF(X). With the development of research, it has shown that 2-ary α-resolution automated reasoning based on lattice-valued logic aiming at processing uncertain information with incomparability is scientific and effective. However, there are some limitations in 2-ary α-resolution automated reasoning, for example, (1) 2-ary α-resolution can only process the resolution of 2-ary generalized literals; (2) the number of resolution generalized literals is fixed at 2 in each resolution. The limitations of these two aspects make the 2-ary α-resolution automated reasoning theory and applications are limited, and also directly affect the efficiency of 2-ary α-resolution automated reasoning. To resolve these limitations, Xu et al. [12] extend the 2-ary α-resolution in this lattice-valued logic into multi-ary α-resolution, and the multi-ary α-resolution principle is introduced in lattice-valued propositional logic LP(X). Multi-ary α-resolution principle provides a new framework for automated reasoning based on lattice-valued logic with truth-value in a LIA. However, it is only a principle not a kind of method. There is no new automated reasoning method under the multi-ary α-resolution principle. Therefore, it is necessary to develop the multi-ary α-resolution methods under the framework of the multi-ary α-resolution principle in order to improve the efficiency of multi-ary α-resolution automated reasoning.

The current paper focuses on a new refinement of multi-ary α-resolution, that is, multi-ary α-semantic resolution, which is a new automated reasoning method based on multi-ary α-resolution principle for lattice-valued logics with truth-value in lattice implication algebras. In Section 2, we mainly list some basic concepts and some properties of lattice implication algebras, lattice-valued propositional logic and lattice valued first order logic, and they will be used in other sections. In Section 3, we mainly investigate the multi-ary α-semantic resolution automated reasoning method based on LP(X) with truth-value in a lattice implication algebra, study the soundness and completeness theorems on this resolution method. In Section 4, we mainly investigate the multi-ary α-semantic resolution automated reasoning algorithm based on LP(X), study the soundness and completeness theorems on this algorithm. In Section 5, conclusions are given. This work will provide a theoretical foundation for the more efficient resolution based automated reasoning in lattice-valued logic.

2. Preliminaries.

In the following, we will introduce some elementary concepts and conclusions of a lattice-valued logic with truth-value in a lattice implication algebra. We refer the readers to [13] for more details.

2.1. Lattice implication algebras.

Definition 2.1. [6] *Let $(L, \vee, \wedge, O, I)$ be a bounded lattice with an order-reversing involution $'$, the greatest element I and the smallest element O, and*

$$\rightarrow: L \times L \longrightarrow L$$

be a mapping. $\mathcal{L} = (L, \vee, \wedge, ', \rightarrow, O, I)$ *is called a lattice implication algebra if the following conditions hold for any* $x, y, z \in L$*:*

(I_1) $x \rightarrow (y \rightarrow z) = y \rightarrow (x \rightarrow z)$*;*
(I_2) $x \rightarrow x = I$*;*
(I_3) $x \rightarrow y = y' \rightarrow x'$*;*
(I_4) $x \rightarrow y = y \rightarrow x = I$ *implies* $x = y$*;*
(I_5) $(x \rightarrow y) \rightarrow y = (y \rightarrow x) \rightarrow x$*;*
(l_1) $(x \vee y) \rightarrow z = (x \rightarrow z) \wedge (y \rightarrow z)$*;*
(l_2) $(x \wedge y) \rightarrow z = (x \rightarrow z) \vee (y \rightarrow z)$*.*

In this paper, we denote $\mathcal{L}$ as a lattice implication algebra $(L, \vee, \wedge, ', \rightarrow, O, I)$.

We list some basic properties of lattice implication algebras. It is useful to develop these topics in other sections.

Example 2.1. *Let* $L = \{O, a, b, c, d, I\}$*, the Hasse diagram of* L *be defined as Figure 1 and its implication operator* $\rightarrow$ *be defined as Table 1 and operator* $'$ *be defined as Table 2. Then* $L = (L, \vee, \wedge, ', \rightarrow, O, I)$ *is a lattice implication algebra.*

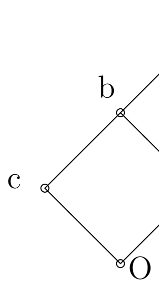

FIGURE 1. Hasse diagram of L

TABLE 1. $\rightarrow$ of $\mathcal{L}$

$\rightarrow$	0	a	b	c	d	1
0	1	1	1	1	1	1
a	c	1	b	c	b	1
b	d	a	1	b	a	1
c	a	a	1	1	a	1
d	b	1	1	b	1	1
1	0	a	b	c	d	1

TABLE 2. $'$ of $\mathcal{L}$

$'$	
O	I
a	c
b	d
c	a
d	b
I	O

Example 2.2. *(Lukasiewicz implication algebra on finite chain) Let* $L_n = \{a_i | i = 1, 2, \cdots, n\}$*,* $a_1 \le a_2 \le \cdots \le a_n$*. For any* $1 \le j, k \le n$*, define*

$$a_j \vee a_k = a_{\max\{j,k\}}, \quad a_j \wedge a_k = a_{\min\{j,k\}},$$

$$(a_j)' = a_{n-j+1}, \quad a_j \rightarrow a_k = a_{\min\{n-j+k,n\}}.$$

Then $(L_n, \vee, \wedge, ', \rightarrow, a_1, a_n)$ *is a lattice implication algebra.*

Theorem 2.1. [9] *Let* $\mathcal{L}$ *be a lattice implication algebra. Then for any* $x, y, z \in L$*, the following conclusions hold:*

(1) if $I \rightarrow x = I$*, then* $x = I$*;*
(2) $I \rightarrow x = x$ *and* $x \rightarrow O = x'$*;*
(3) $O \rightarrow x = I$ *and* $x \rightarrow I = I$*;*
(4) $(x \rightarrow y) \rightarrow ((y \rightarrow z) \rightarrow (x \rightarrow z)) = I$*;*

(5) $(x \to y) \vee (y \to x) = I$;
(6) if $x \le y$, then $x \to z \ge y \to z$ and $z \to x \le z \to y$;
(7) $x \le y$ if and only if $x \to y = I$;
(8) $(z \to x) \to (z \to y) = (x \wedge z) \to y = (x \to z) \to (x \to y)$;
(9) $x \to (y \vee z) = (y \to z) \to (x \to z)$;
(10) $x \to (y \to z) = (x \vee y) \to z$ if and only if $x \to (y \to z) = x \to z = y \to z$;
(11) $z \le y \to x$ if and only if $y \le z \to x$.

2.2. **Multi-ary α-resolution principle based on lattice-valued propositional logic LP(X).** In this section, we will list multi-ary α-resolution principle for lattice-valued logics with truth-value in lattice implication algebras, and it will be used on Section 3.

Definition 2.2. [2] *Let X be a set of propositional variables, $T = L \cup \{', \to\}$ be a type with $ar(') = 1$, $ar(\to) = 2$ and $ar(\alpha) = 0$ for any $\alpha \in L$. The propositional algebra of the lattice-valued propositional calculus on the set X of propositional variables is the free T-algebra on X denoted by LP(X).*

Theorem 2.2. [3] *LP(X) is the minimal set Y which satisfies:*
(1) $X \cup L \subseteq Y$;
(2) if $p, q \in Y$, then $p', p \to q \in Y$.

Remark 2.1. *In a lattice implication algebra $\mathcal{L}$, for any $\alpha, \beta \in L$,*

$$\alpha \vee \beta = (\alpha \to \beta) \to \beta,$$
$$\alpha \wedge \beta = (\alpha' \vee \beta')'.$$

Hence, $\mathcal{L}$ and LP(X) can be looked at algebras with the same type $T = \mathcal{L} \cup \{', \to\}$ and for any $p, q \in \mathcal{F}$,

$$p \vee q = (p \to q) \to q,$$
$$p \wedge q = (p' \vee q')'.$$

Definition 2.3. [2] *A valuation of LP(X) is a propositional algebra homomorphism v : $LP(X) \to L$.*

Definition 2.4. [6] *Let $p \in LP(X)$, $\alpha \in L$. If there exists a valuation v of LP(X) such that $v(p) \not\le \alpha$, p is satisfiable by a truth-value level α, in short, α-satisfiable; if $v(p) \not\le \alpha$ for every valuation v, p is valid by the truth-value level α, in short, α-valid. If $\alpha = I$, then p is valid simply.*

Definition 2.5. [9] *Let $p \in LP(X)$. If $v(p) \le \alpha$ for any valuation v of LP(X), p is always false by the truth-valued level α, in short, α-false. If $\alpha = O$, then p is false.*

Definition 2.6. [12] (**Multi-ary α-Resolution Principle**) *Let $C_i = p_{i1} \vee \cdots \vee p_{i m_i}$ be generalized clauses of LP(X), $H_i = \{p_{i1}, \cdots, p_{i m_i}\}$ the set of all disjuncts occurring in C_i, $i = 1, 2, \cdots, m$, $\alpha \in L$. For any $i \in \{1, 2, \cdots, m\}$, if there exist generalized literals $x_i \in H_i$ such that $x_1 \wedge x_2 \wedge \cdots \wedge x_m \le \alpha$, then*

$$C_1(x_1 = \alpha) \vee C_2(x_2 = \alpha) \vee \cdots \vee C_m(x_m = \alpha)$$

is called an m-ary α-resolvent of $C_1, C_2, \cdots, C_m$, denoted by

$$R_{p(m-\alpha)}(C_1(x_1), C_2(x_2), \cdots, C_m(x_m)),$$

x_1, $x_2, \cdots, x_m$ are called an m-ary α-resolution group. The m-ary α-resolution group $x_1, x_2, \cdots, x_m$, is denoted by $(x_1, x_2, \cdots, x_m) - \alpha$.

Remark 2.2. *In Definition 2.6, the symbol $C_i(x_i = \alpha)$ is obtained by replacing x_i with α in the generalized clause C_i, $i = 1, 2, \cdots, m$.*

Definition 2.7. [12] *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in lattice-valued propositional logic LP(X) and $\alpha \in L$. A sequence:*

$$\Phi_1, \Phi_2, \cdots, \Phi_t$$

is called a multi-ary α-resolution deduction from S to Φ_t, if it satisfies the following conditions:

(1) $\Phi_i \in \{C_1, C_2, \cdots, C_m\}$ $(i = 1, 2, \cdots, t)$ or

(2) Φ_i is a multi-ary α-resolvent.

Theorem 2.3. [12] (**Soundness**) *Suppose $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in LP(X). $\{\Phi_1, \Phi_2, \cdots, \Phi_t\}$ is a multi-ary α-resolution deduction from S to Φ_t. If $\Phi_t \leq \alpha$, then $S \leq \alpha$.*

Theorem 2.4. [12] (**Completeness**) *Suppose $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in LP(X). If $S \leq \alpha$, then there exists a multi-ary α-resolution deduction from S to α-empty clause.*

In this paper, α is assumed to be always less than I.

3. Multi-Ary α-Semantic Resolution Method Based on LP(X).

In this section, the multi-ary α-semantic resolution method in lattice-valued propositional logic LP(X) will be investigated based on the multi-ary α-resolution principle which has been listed in Section 2. The soundness and completeness of this method are also given in this section.

Definition 3.1. *Let v be a valuation in lattice-valued propositional logic LP(X), $\alpha \in L$. $N, E_1, \cdots, E_q$ are generalized clauses sets in LP(X), and $\mathcal{G}$ is an order of generalized literals occurring in these clauses. The finite sequence $(N, E_1, E_2, \cdots, E_q)(*)$ is called a multi-ary α-semantic clash w.r.t. v and $\mathcal{G}$, if $(*)$ satisfy the following conditions:*

(1) For any generalized clause $C \in E_i$, $v(C) \leq \alpha$, $i = 1, 2, \cdots, q$;

(2) Let $R_0 = \vee_{G_j \in N}(G_j)$, for any $i = 1, 2, \cdots, q$, there exists a multi-ary α-resolution formula R_i of N_i and E_i, where $\phi \neq N_1 \subseteq N$, $N_i = \{R_1\} \cup N_2^*$, $N_2^* \subset N$. For any $i = 3, 4, \cdots, q$,

$$N_i = \{R_{i-1}\} \cup N_i^*,$$
$$N_i^* \subset N \cup \{R_1, R_2, \cdots, R_{i-2}\};$$

(3) For any generalized clause $C \in E_i$, the α-resolution generalized literals in C is the leftmost generalized literals in C;

(4) $v(R_q) \leq \alpha$.

R_q is called multi-ary α-sematic resolvent of this clash w.r.c. v and $\mathcal{G}$, N is called the core and

$$E_1, E_2, \cdots, E_q$$

are called α-electrons group.

Remark 3.1. *(1) In this definition, for any generalized clause C, if the same disjunctive terms of C occur in different places in C, then the leftmost disjunction should be reserved and others should be deleted.*

(2) In this definition, there exists $G \in N$ such that G must be α-true, that is, $v(G) \nleq \alpha$. In fact, if $v(G) \leq \alpha$ for any $G \in N$, then $v(R_0) \leq \alpha$, i.e., there is not a multi-ary α-semantic clash. If the R_0 is regarded as the multi-ary α-resolvent w.r.t. v and $\mathcal{G}$, then the multi-ary α-semantic resolvent w.r.t. v and $\mathcal{G}$ will be redundancy.

(3) For any disjunctive term g in E_i $(i = 1, 2, \cdots, q)$, $v(g) \leq \alpha$.

(4) In a multi-ary α-semantic clash, for the ith multi-ary α-semantic clash, the resolvent R_{i-1} must occur in the N_{i-1}. However, the generalized clauses that resolve with R_{i-1} may appear in $N, R_1, \cdots, R_{i-2}$ besides E_i. Therefore,

$$\begin{aligned} N_i &= \{R_{i-1}\} \cup N_i^*, \\ N_i^* &\subset N \cup \{R_1, R_2, \cdots, R_{i-2}\}. \end{aligned}$$

Example 3.1. *In lattice-valued propositional logic $L_9P(X)$, let $\alpha = a_6$, generalized clause set $S = \{C_1, C_2, C_3, C_4\}$, where*

$$\begin{aligned} C_1 &= (x \to y), \\ C_2 &= (x \to z)' \vee (s \to t), \\ C_3 &= (y \to z)' \vee (y \to a_2), \\ C_4 &= (s \to t)' \vee (s \to q) \end{aligned}$$

where $a_2 \in L_9$, x, y, z, s, t are propositional variables. Define a valuation v of $L_9P(X)$ as follows:

$$v(x) = I, \quad v(y) = a_7, \quad v(z) = a_3, \quad v(s) = v(t) = a_5,$$

then $v(C_1) > \alpha$, $v(C_2) > \alpha$, $v(C_3) < \alpha$, $v(C_4) < \alpha$.

Let $\mathcal{G} : (s \to t)', (y \to z)', (x \to z)', x \to y, y \to a_2, s \to t$ be an order of generalized literal in C_1, C_2, C_2, C_4.

As

$$N_1 = \{C_2\} \subseteq N = \{C_1, C_2\}, \quad E_1 = \{C_4\},$$

we have $R_1 = (x \to z)' \vee \alpha$.

$$N_2 = \{R_1\} \cup N_2^* = \{R_1, C_1\}, \quad E_2 = \{C_3\},$$

we have

$$R_2 = (y \to a_2) \vee \alpha,$$

where $N_2^* = \{C_1\} \subseteq N = \{C_1, C_2\}$.

As $v(R_2) \le \alpha$, (E, R_1, R_2) is a multi-ary α-semantic clash w.r.t v and $\mathcal{G}$. $(E_1 = \{C_3\}$, $E_2 = \{C_4\})$ is α-electrons group and $N = \{C_1, C_2\}$ is the α-core of this clash.

Remark 3.2. *In Example 3.1, for the generalized literals $y \to z$, there does not exist 2-ary α-resolute group, so there is not a 2-ary α-semantic resolvent. However, there exists 3-ary α-semantic resolute group $x \to y, y \to z, (x \to z)'$.*

Example 3.2. *Let $L_6P(X)$ be a lattice-valued propositional logic, whose truth in a lattice implication algebra listed in* **Example 2.1**. *Let $\alpha = b$, generalized clause set $S = \{C_1, C_2, C_3\}$, where*

$$\begin{aligned} C_1 &= (x \to y) \vee (s \to t), \\ C_2 &= (y \to z) \vee (w \to t) \vee (p \to q), \\ C_3 &= (x \to z)' \vee (s \to q). \end{aligned}$$

Define a valuation v of $L_6P(X)$ as follows:

$$v(x) = v(y) = v(s) = v(w) = v(p) = I,$$

$$v(z) = v(t) = v(q) = d,$$

then $v(C_1) > \alpha$, $v(C_2) = d \le \alpha$, $v(C_3) = b \le \alpha$.

Let $\mathcal{G} : y \to z, (x \to z)', x \to y, s \to t, w \to t, p \to q, s \to q$ be an order of generalized literal in C_1, C_2, C_3 and $R_0 = \{C_1\}$. Since the leftmost generalized literals are $y \to z, (x \to z)'$ in E_1, E_2, respectively, and

$$(x \to y) \wedge (y \to z) \wedge (x \to z)' \leq \alpha,$$

we have

$$R_1(C_1, C_2, C_3) = \alpha \vee (s \to t) \vee (w \to t) \vee (p \to q) \vee (s \to q)$$

and

$$v(R_1(N, E_1, E_2)) = v(\alpha \vee (s \to t) \vee (w \to t) \vee (p \to q) \vee (s \to q)) = a_6 \leq \alpha.$$

Therefore, $(\{C_1\}, \{C_2, C_3\})$ is a multi-ary α-semantic semantic clash w.r.t. v and $\mathcal{G}$, $\{C_2, C_3\}$ is an α-electron group and $N = \{C_1\}$ is an α-core of this clash, R_1 is a multi-ary α-semantic resolvent w.r.t. v and $\mathcal{G}$.

Remark 3.3. *We change the order of generalized literals in the generalized clause set S, give another order $\mathcal{G}_1 : y \to z, w \to t, s \to q, (x \to z)', x \to y, s \to t, p \to q$, then there will not exist a multi-ary α-semantic clash with respect to v and $\mathcal{G}_1$. This shows that determination of the order on generalized literals is very important in the multi-ary α-semantic clash.*

Theorem 3.1. *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where*

$$C_1, C_2, \cdots, C_m$$

are generalized clauses in lattice-valued propositional logic LP(X), v be a valuation in LP(X) and $\alpha \in L$, $\mathcal{G}$ is an order of generalized literals occurring in these clauses. If there exists a multi-ary α-semantic clash w.r.t. v and $\mathcal{G}$, R_s is a multi-ary α-semantic resolvent of this clash, then

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m \leq R_s.$$

Proof: The proof is straightforward from [12].

Definition 3.2. *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in lattice-valued propositional logic LP(X), v be a valuation in LP(X) and $\alpha \in L$, $\mathcal{G}$ is an order of generalized literals occurring in these clauses. A sequence:*

$$\Phi_1, \Phi_2, \cdots, \Phi_t$$

is called a multi-ary α-semantic resolution deduction from S to Φ_t, if it satisfies the following conditions:

(1) $\Phi_i \in \{C_1, C_2, \cdots, C_m\}$ $(i = 1, 2, \cdots, t)$ or

(2) Φ_i is a multi-ary α-semantic resolvent w.r.t. v and $\mathcal{G}$, where the core and electrons of Φ_i are composed of Φ_j $(j < i)$ or generalized clauses occurring in S.

Theorem 3.2. *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in lattice-valued propositional logic LP(X), v be a valuation in LP(X) and $\alpha \in L$, $\mathcal{G}$ is an order of generalized literals occurring in these clauses. There exists an α-$\mathcal{G}v$ resolution deduction from S to Φ_t:*

$$\Phi_1, \Phi_2, \cdots, \Phi_t,$$

and Φ_t is α-empty clause, then $S \leq \alpha$.

Proof: According to the soundness of the general form of α-resolution principle in lattice-valued propositional logic LP(X), we can obtain the result easily.

Theorem 3.3. (**Condition Completeness**) *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in lattice-valued propositional logic LP(X), v be a valuation in LP(X) and $\alpha \in L$, $\mathcal{G}$ is an order of generalized literals occurring in these clauses. If the following conditions hold:*

(1) $S \leq \alpha$;

(2) $S_1 = \{C_i | v(C_i) \leq \alpha\} \neq \emptyset$ and $S_2 = \{C_j | v(C_j) \nleq \alpha, i \in \{1, 2, \cdots, m\}\} \neq \emptyset$;

then there exists a multi-ary α-semantic resolution deduction from S to α-empty clause.

Proof: The proof of this theorem includes the following two cases:

1). There exists α-false generalized clause C_p in S.

Assume $C_p = g_1 \vee g_2 \vee \cdots \vee g_u \leq \alpha$, then

$$g_t \leq \alpha, \ \ t = 1, 2, \cdots, u, \ \ p \in \{1, 2, \cdots, m\}.$$

Since $S_2 \neq \emptyset$, then there exist $j \in \{1, 2, \cdots, m\}$,

$$C_j = h_1 \vee h_2 \vee \cdots \vee h_w$$

such that $v(C_j) \nleq \alpha$, then there exists at least one h_y such that $v(h_y) \nleq \alpha$ for $y \in \{1, 2, \cdots, w\}$. Since $g_t \wedge h_y \leq \alpha$ for any $t = 1, 2, \cdots, u$, $y \in \{1, 2, \cdots, w\}$, then there exists a multi-ary α-semantic clash

$$(N', E'_1, \cdots, E'_w),$$

where $N' = \{C_j\}$, $E'_1 = \cdots = E'_w = \{C_p\}$, the multi-ary α-resolvent R^1_w of this clash is obtained by replacing leftmost generalized literal of C_p occurring in $\mathcal{G}$ with α, then $R^1_w \leq \alpha$. At the same time, there exists the second multi-ary α-semantic clash

$$\left(N^2, R^1_w, \cdots, R^1_w\right),$$

where $N^2 = \{C_j\}$. In this clash, the multi-ary α-resolvent R^2_w of this clash is obtained by replacing leftmost generalized literal of R^1_w occurring in $\mathcal{G}$ with α, then $R^2_w \leq \alpha$. According to this way, we have $R^u_w \leq \alpha$ for the number of disjunctive term of C_p is finite. Therefore, theorem holds under this situation.

2). There is no α-false generalized clause in S.

Let H_i be the set composed of all generalized literals occurring in C_i and $|H_i| = w_i$, where $i = 1, 2, \cdots, m$. Suppose $K(S)$ is equal to the difference of the number of generalized literals from that of generalized clauses occurring in S, i.e., $K(S) = \sum_i^m w_i - m$. Two cases need to be discussed.

1° If $K(S) = 0$, S is composed of unit generalized clauses, i.e., each generalized clause only containing one generalized literal. By condition (1), we have $S \leq \alpha$; therefore, all generalized literal is a multi-ary α-resolution group, and so there is a multi-ary α-semantic clash, the nuclear of this clash is $\{C_1, C_2, \cdots, C_m\} \setminus S_1$ and the electronic is S_1, and obviously, the multi-ary α-semantic resolvent is α-false. So Theorem 3.3 holds under the situation $K(S) = 0$.

2° Suppose that the result holds for $K(S) < n$ $(n > 0)$. Now we need to prove the result also holds for $K(S) = n$.

Let $K(S) = n$ $(n > 0)$, then S has at least one non-unit generalized clause in S. Let C_i be a non-unit generalized clause in S and H be a set of all generalized literals occurring in all non-unit generalized clauses.

(**A**) If there exists $g \in H$ such that $v(g) \leq \alpha$, assume that $C_i = C_i^* \vee g$, where C_i is a nonempty generalized clause. We define the following generalized clause set as follows:

$$S_3 = \{S - C_i\} \cup \{C_1^*\}.$$

As $S \leq \alpha$, $S_3 \leq \alpha$ and $K(S_3) < n$. By the induction hypothesis, there exists a multi-ary α-semantic resolution deduction D_2 from S_3 to α-empty clause.

For any multi-ary α-semantic clash

$$\left(N^2, E_1^2, \cdots, E_s^2\right)$$

in D_2. Let R_s^2 be a multi-ary α-semantic resolvent of this clash. Three cases need to be discussed.

(**Case I:**) If C_i^* is an element occurring in the core of each multi-ary α-senmatic clash

$$\left(N^2, E_1^2, \cdots, E_s^2\right)$$

of D_2, then D_2 can be amended as

$$\left(N^{2*}, E_1^2, \cdots, E_s^2\right)$$

and its multi-ary α-semantic resolvent is equal to $R_s^2 \vee g$, where N^{2*} is obtained by replacing C_i^* occurring in N^2 with $C_i^* \vee g$, and R_s^2 is the multi-ary α-semantic resolvent of clash $(N^2, E_1^2, \cdots, E_s^2)$.

(**Case II:**) If C_i^* is an element of electrons in the multi-ary α-senmatic clash $(N^2, E_1^2, \cdots, E_s^2)$, then there exist $j \in \{1, 2, \cdots, s\}$ such that $C_i^* \in E_j^2$, we replace C_i^* with C_i in this clash. Let E_j^{2*} be the set obtained by replacing C_i^* occurring in E_j^2 with C_i, then we obtain a new sequence

$$\left(N^2, E_1^2, E_2^2, \cdots, E_{j-1}^2, E_j^{2*}, E_{j+1}^2, \cdots, E_s^2\right).$$

And the sequence is also a multi-ary α-semantic clash and the multi-ary α-semantic resolvent is $R_s^2 \vee g$.

(**Case III:**) The electronics of $(N^2, E_1^2, \cdots, E_s^2)$ contains a multi-ary α-semantic resolvent R^0, where R^0 is generated by a multi-ary α-semantic clash containing C_i^* as an element of electronic. Without loss of generality, we can assume $R^0 \in E_j^2$ is a multi-ary α-semantic resolvent, C_i^* is an element of electronic in the multi-ary α-semantic clash generating R^0, where $j \in \{1, 2, \cdots, s\}$. As the disjunctions, in multi-ary α-semantic resolvent which composed of some disjunctions of non-α-resoluted generalized literals in non-unit generated clauses of α-electronic group, are α-false under the valuation v in α-core of this multi-ary α-semantic clash. The sequence

$$\left(N^2, E_1^2, E_2^2, \cdots, E_{j-1}^2, E_j^{2*}, E_{j+1}^2, \cdots, E_s^2\right)$$

is also a multi-ary α-semantic clash and its multi-ary α-semantic resolvent is equal to $R_s^2 \vee g$, where E_j^{2*} is the set obtained by replacing R^0 occurring in E_j^2 with $R_s^2 \vee g$, and R_s^2 is the multi-ary α-semantic resolvent of clash $(N^2, E_1^2, \cdots, E_s^2)$.

Therefore, we can replace C_i^* occurring in any multi-ary α-semantic clash of D_2 with C_i and modifying the corresponding multi-ary α-semantic resolvent, we can obtain a resolution deduction D_{21} from S to α-empty clause or g.

If D_{21} is a multi-ary α-semantic resolution deduction from S to α-empty clause, then the conclusion holds.

If D_{21} is a multi-ary α-semantic resolution deduction from S to g, then we consider clause set $S_5 = S \cup \{g\}$, $S_5 \leq \alpha$ and $\{g\}$ is a unit α-false generalized clause. By **Case I**, we can get a multi-ary α-semantic resolution deduction D_{22} from S_5 to α-empty clause, and connecting D_{21} and D_{22}, we can get a multi-ary α-semantic resolution deduction D from S to α-empty clause.

(**B**) For any $g \in H$ such that $v(g) \nleq \alpha$. We have $v(C) \nleq \alpha$ for any non-unit generalized clause of S. As $S_1 \neq \emptyset$ and any generalized clause of S_1 are all α-false under valuation v, all generalized clauses in S_1 are all unit generalized clauses. So

$$g_1 \wedge g_2 \wedge \cdots \wedge g_m \leq \alpha$$

for any $g_i \in H_i$, where $i = 1, 2, \cdots, m$. Then there exists a muti-ary α-semantic clash whose α-cores are composed of the generalized clause in $S \backslash S_1$ and the α-electronics are composed of the generalized clause in S_1, the multi-ary α-semantic resolvent of this clash is α-empty clause. Therefore, Theorem 3.3 is valid.

Theorem 3.4. *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, C_2, \cdots, C_m$ are generalized clauses in lattice-valued propositional logic $L_n P(X)$, v be a valuation in $L_n P(X)$ and $\alpha \in L$, $\mathcal{G}$ is an order of generalized literals occurring in these clauses. If the following conditions hold:*

(1) $S \leq \alpha$;

(2) There exists a generalized clause C_j such that for any disjunctive term g in C_j, $v(g) > \alpha$, where $j \in \{1, 2, \cdots, m\}$;

then there exists a multi-ary α-semantic resolution deduction from S to α-empty clause.

Proof: From condition (2), we have $S_2 \neq \emptyset$. Since $S \leq \alpha$ and $\alpha \in L_n$, then there exist

$$C_j \in \{C_1, C_2, \cdots, C_m\}$$

such that $v(C_j) \leq \alpha$. Hence $S_1 \neq \emptyset$. It follows from Theorem 3.3 that Theorem 3.4 holds.

Example 3.3. *Let*

$$\begin{aligned} C_1 &= x \to y, \\ C_2 &= (x \to z)' \vee (s \to t), \\ C_3 &= (y \to z) \vee (y \to a_2) \vee (a_5 \to q), \\ C_4 &= (s \to t)', \\ C_5 &= (p \to q)' \end{aligned}$$

be five generalized clauses in lattice-valued propositional logic $L_9 P(X)$ and $S = C_1 \wedge C_2 \wedge \cdots \wedge C_5$, where $a_2, a_5 \in L_9$, x, y, z, s, t, p, q are propositional variables. Let $\alpha = a_6$ and v be a valuation in $L_9 P(X)$ such that

$$v(x) = I, \quad v(y) = a_7, \quad v(z) = a_3,$$
$$v(s) = v(t) = v(p) = a_5, \quad v(q) = I,$$

then

$$v(C_1) > \alpha, \quad v(C_2) > \alpha, \quad v(C_3) > \alpha, \quad v(C_4) < \alpha, \quad v(C_5) < \alpha.$$

Let $\mathcal{G}$ be an order of generalized literals, where

$$\mathcal{G} : (s \to t)', (p \to q)', y \to z, (x \to z)', x \to y, y \to a_2, s \to t, a_5 \to q.$$

Then there exists the following multi-ary α-semantic resolution deduction ω from S to α-empty clause:

(1) $x \to y$

(2) $(x \to z)' \vee (s \to t)$

(3) $(y \to z) \vee (y \to a_2) \vee (a_5 \to q)$

(4) $(s \to t)'$

(5) $(p \to q)'$

(6) $(y \to z) \vee (y \to a_2) \vee \alpha$ *by* (3) (5)

(7) $(y \to a_2) \vee \alpha$ *by* (1) (2) (4) (6)

(8) *α-empty clause* *by* (1) (2) (4) (7)

In fact, there are three multi-ary α-semantic clashes in ω:

(1) $N_1^1 = \{C_3\}$, $E_1^1 = \{C_5\}$, *the resolvent* R_1^1 *of clash* (N_1^2, E_1^2) *is*

$$(y \to z) \vee (y \to a_2) \vee \alpha;$$

(2) $N_1^2 = \{C_1, C_2\}$, $E_1^2 = \{R_1^1\}$, $E_2^2 = \{C_4\}$, *the resolvent* R_2^2 *of clash* (N_1^3, E_1^2, E_2^2) *is* $(y \to a_2) \vee \alpha$, *where*

$$R_2^1 = (x \to z)' \vee \alpha, N_1^1 = \left\{R_1^1\right\};$$

(3) $N_1^3 = \{C_1, C_2\}$, $E_1^3 = \{C_4\}$, $E_2^3 = \{R_2^2\}$, *the resolvent of clash* (N_1^3, E_1^3, E_2^3) *is* α*-empty clause.*

Remark 3.4. *According to the 2-ary α-resolution principle, the generalized clause (8) occurring in deduction ω does not have a 2-ary α-resolution pair. So there does not exist a 2-ary α-semantic resolution deduction from S to α-empty clause.*

Remark 3.5. *From Example 3.3, the number of generalized literals taking part in α-resolute in each muti-ary α-semantic clash is not fixed, and this reflects the multi-ary α-semantic resolution deduction is dynamic. The dynamic of resolution deduction demonstrates the high efficiency of multi-ary α-semantic resolution automated reasoning.*

4. **Realization for Multi-Ary α-Semantic Resolution Method Based on LP(X).** In this section, we will construct the algorithm for multi-ary α-semantic resolution methods. Without loss of generality, we assume that $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ is a generalized clause set in a lattice propositional logical system LP(X), where $C_1, C_2, \cdots, C_m$ are generalized clauses of LP(X). In this section, let $\alpha \in L$ and α is dual molecules. We pretreat the generalized clause set before the specific algorithm is given, and the concrete steps are as follows.

Step 1: If the sets S_1, S_2 in Theorem 3.3 are nonempty under the valuation v of LP(X), go to Step 2; otherwise, the multi-ary α-semantic resolution methods are not suitable for generalized clause set S.

Step 2: Check all generalized clauses occurring in S: If there exists generalized clause $C_k \leq \alpha$, then pretreatment stops and $S \leq \alpha$. Otherwise, go to Step 3.

Step 3: Check any disjunctive term g occurring in S: If $g \leq \alpha$, then delete g, the pretreatment stops.

Theorem 4.1. *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a generalized clause set in LP(X) and $\alpha \in L$. S^* is generalized clause set obtained by pretreating S, then $S \leq \alpha$ if and only if $S^* \leq \alpha$.*

Proof: Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m \leq \alpha$ and v be valuation of lattice-valued propositional logic LP(X). If there exists a disjunctive term $g_i \in C_i$ $(i \in \{1, 2, \cdots, m\})$ such that $g_i \leq \alpha$. We transform S into a generalized disjunctive normal form. As $S \leq \alpha$, very disjunctive term in generalized disjunctive normal forms is α-false. Therefore, the generalized disjunctive norm forms obtained by deleting disjunctive term containing g_i are still α-false, so $S^* \leq \alpha$.

Conversely, let $S^* = C_1^* \wedge C_2^* \wedge \cdots \wedge C_m^* \leq \alpha$, S^* is a generalized clause set obtained after pretreating S. Therefore, there exists $C_i^* \in S^*$ such that C_i^* is obtained by deleting α-false disjunctive term g from C_i, that is, $C_i^* \vee g = C_i$. Therefore,

$$\begin{aligned}
S^* &= C_1^* \wedge C_2^* \wedge \cdots \wedge C_{i-1}^* \wedge C_i \wedge C_{i+1}^* \wedge \cdots \wedge C_m^* \\
&= C_1^* \wedge C_2^* \wedge \cdots \wedge C_{i-1}^* \wedge (C_i \vee g) \wedge C_{i+1}^* \wedge \cdots \wedge C_m^* \\
&= \left(C_1^* \wedge C_2^* \wedge \cdots \wedge C_{i-1}^* \wedge C_i^* \wedge C_{i+1}^* \wedge \cdots \wedge C_m^*\right) \\
&\vee \left(C_1^* \wedge C_2^* \wedge \cdots \wedge C_{i-1}^* \wedge g \wedge C_{i+1}^* \wedge \cdots \wedge C_m^*\right) \\
&\leq S^* \vee \left(C_1^* \wedge C_2^* \wedge \cdots \wedge C_{i-1}^* \wedge g \wedge C_{i+1}^* \wedge \cdots \wedge C_m^*\right) \leq \alpha.
\end{aligned}$$

Now we give an algorithm for multi-ary α-semantic resolution methods based on LP(X). In this algorithm, we assume that S is a generalized clause set after above pretreatment. The resoluable generalized literals satisfy condition (3) in Definition 3.1.

Step 0: Determine a valuation v of generalized clause, an order $\mathcal{G}$ of generalized literals in lattice-valued propositional logic LP(X) and set $M = \{C \in S|v(C) \leq \alpha\}$, $N = \{C \in S|v(C) \nleq \alpha\}$. If $M, N \neq \emptyset$, turn to Step 1; otherwise algorithm stops and S cannot be resolved by multi-ary α-semantic resolution methods.

Step 1: Set $j = 1$;

Step 2: Put $A_0 = \emptyset$, $B_0 = N$;

Step 3: Set $i = 0$;

Step 4: If A_i contains an α-empty clause, then algorithm stops and $S \leq \alpha$; otherwise, turn to the next step;

Step 5: If $B_i = \emptyset$, then turn to Step 9; otherwise, turn to the next step;

Step 6: Computing the multi-ary α-resolvent of S_1, S_2 satisfies condition (3) in Definition 3.1, where $S_1 \subseteq M$, $S_2 \subseteq B_0 \cup B_1 \cup \cdots \cup B_{i-1}$; Denote the set of all multi-ary α-resolvent as W_{i+1}. If W_{i+1} contains an α-empty clause, then algorithm stops; otherwise, turn to the next step;

Step 7: Let $A_{i+1} = \{\Phi \in W_{i+1}|v(\Phi) \leq \alpha\}$, $B_{i+1} = \{\Phi \in W_{i+1}|v(\Phi) \nleq \alpha\}$; If $A_{i+1} = \emptyset$, then turn to the next step; otherwise, turn to Step 9;

Step 8: Set $i = i + 1$, turn to Step 4;

Step 9: Put $T = A_0 \cup A_1 \cup \cdots \cup A_i$, $M = M \cup T$;

Step 10: Set $j = j + 1$;

Step 11: Computing the multi-ary α-resolvent of S_1, S_2 satisfies condition (3) in Definition 3.1, where $S_1 \subseteq T$, $S_2 \subseteq N$; Denote the set of all multi-ary α-resolvent as $\mathcal{R}$. If $\mathcal{R}$ contains an α-empty clause, then algorithm stops; otherwise, turn to the next step;

Step 12: Let $A_{i+1} = \{\Phi \in \mathcal{R}|v(\Phi) \leq \alpha\}$; $B_{i+1} = \{\Phi \in \mathcal{R}|v(\Phi) \nleq \alpha\}$; Turn to Step 3.

Theorem 4.2. (**Soundness**) *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a generalized clause set in lattice-valued propositional logic LP(X) and $\alpha \in L$, applying above algorithm on S. If the algorithm terminates in Step 4, then $S \leq \alpha$.*

Proof: If the algorithm terminates in Step 4, then there exists a multi-ary α-semantic resolution deduction from S to α-empty clause. It follows from Theorem 3.2 that $S \leq \alpha$.

Theorem 4.3. (**Completeness**) *Let $S = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a generalized clause set in lattice-valued propositional logic LP(X) and $\alpha \in L$, applying above algorithm on S. If $S \leq \alpha$, then the algorithm terminates Step 4.*

Proof: If S contains α-empty clause, then the α-empty clause must be in M. When $i = 0$, we can choose some generalized clauses in M and B_0 (i.e., N), respectively, to take part in the multi-ary α-resolution. The multi-ary α-resolvents and α-empty clauses will be in W_1. It follows from Step 7 that α-empty clauses will be in A_1; therefore, the algorithm will terminate Step 4.

If S does not contain α-empty clause, the algorithm cannot cycle infinitely. If the algorithm loops infinitely to loop variable i, that is, $B_i \neq \emptyset$, there is always multi-ary α-resolvent which is not less than or not equal to α under the valuation v in the multi-aryα-resolution deduction. As N is finite, this case is impossible. If the algorithm loops infinitely to loop variable j, then there is no α-empty clause in the multi-ary α-resolution deduction, which contradicts with $S \leq \alpha$. Therefore, the algorithm cannot loop infinitely, and it must be terminated in Step 4.

Now, we can show the validity of the algorithm by an example.

Example 4.1. *Let*

$$\begin{aligned} C_1 &= x \to y, \\ C_2 &= (x \to z)' \vee (s \to t), \\ C_3 &= (y \to z) \vee (y \to a_2) \vee (a_5 \to q), \\ C_4 &= (s \to t)', \\ C_5 &= (p \to q)' \end{aligned}$$

be five generalized clauses in lattice-valued propositional logic $L_9P(X)$ and $S = C_1 \wedge C_2 \wedge \cdots \wedge C_5$, where $a_2, a_5 \in L_9$ and x, y, z, s, t, p, q are propositional variables. Let $\alpha = a_6$ and v be a valuation in $L_9P(X)$ such that

$$\begin{aligned} &v(x) = I, \quad v(y) = a_7, \quad v(z) = a_3, \\ &v(s) = v(t) = v(p) = a_5, \quad v(q) = I, \end{aligned}$$

then

$$v(C_1) > \alpha, \quad v(C_2) > \alpha, \quad v(C_3) > \alpha, \quad v(C_4) < \alpha, \quad v(C_5) < \alpha.$$

Let

$$\mathcal{G} : (s \to t)', (p \to q)', y \to z, (x \to z)', x \to y, y \to a_2, s \to t, a_5 \to q$$

be an order of generalized literals in S.

$$\begin{aligned} M &= \{C_4, C_5\} \\ N &= \{C_1, C_2, C_3\} \end{aligned}$$

$$\begin{aligned}
j = 1 :& A_0 = \emptyset, B_0 = N \\
& W_1 = \{(x \to z)' \vee \alpha, (y \to z) \vee (y \to a_2) \vee \alpha\} \\
& A_1 = \{(y \to z) \vee (y \to a_2) \vee \alpha\} \\
& B_1 = \{(x \to z)' \vee \alpha\} \\
& T = \{(y \to z) \vee (y \to a_2) \vee \alpha\} \\
& M = \{C_4, C_5, (y \to z) \vee (y \to a_2) \vee \alpha\} \\
j = 2 :& \mathcal{R} = \{(y \to a_2) \vee (s \to t) \vee \alpha\} \\
& A_0 = \emptyset \\
& B_0 = \{(y \to a_2) \vee (s \to t) \vee \alpha\} \\
& W_1 = \{(y \to a_2) \vee \alpha\} \\
& A_1 = \{(y \to a_2) \vee \alpha\} \\
& B_1 = \emptyset \\
& T = \{(y \to a_2) \vee \alpha\} \\
& M = \{C_4, C_5, (y \to z) \vee (y \to a_2) \vee \alpha, (y \to a_2) \vee \alpha\} \\
j = 3 :& \mathcal{R} = \{(s \to t) \vee \alpha\} \\
& A_0 = \emptyset \\
& B_0 = \{(s \to t) \vee \alpha\} \\
& W_1 = \{\alpha\} \\
& A_1 = \{\alpha\}
\end{aligned}$$

From above Example 4.1, we can obtain three multi-ary α-semantic resolvents

$$(\alpha, (y \to a_2) \vee \alpha, (y \to z) \vee (y \to a_2) \vee \alpha)$$

by applying multi-ary α-semantic resolution algorithm. The result is consistent with Example 3.3 by using multi-ary α-semantic resolution method directly.

5. **Conclusions.** In the previous works [7, 8], α-resolution principle based on lattice-valued logic with truth-value in a lattice implication algebra is carried out through finding α-resolution pairs. As an α-resolution pair only includes two generalized literals, the application of α-resolution principle is limited to a certain extent. By extending α-resolution pairs to α-resolution groups, which can have more than two generalized literals (in fact, it is general case in LP(X)), Xu et al. [12] proposed multi-ary α-resolution principle based on the above lattice-valued logic.

In current paper, we mainly investigated multi-ary α-semantic resolution automated reasoning method based on the multi-ary α-resolution principle for lattice-valued logic with truth-value in a lattice implication algebra. The definitions of the multi-ary α-semantic resolution and multi-ary α-semantic resolution deduction are given, and the soundness and completeness are gotten. The multi-ary α-semantic resolution automated reasoning algorithm along with soundness and completeness is constructed. This will become the theoretical foundation for establishing the resolution method and technique with the goal of applying to some practical fields such as expert system design, intelligent robot design, and machine learning system design.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61175055, 61305074); Science and Technology Projects of Sichuan Province (2015JY0120); The Scientific Research Project of Department of Education of Sichuan Province (14ZA0245, 15T- D0027, 15ZB0270); The Opening Project of Key Laboratory of Nondestructive Testing and Engineering Calculation of Bridge in Sichuan Province (2014QYJ02); The Scientific Research Fund of Neijiang Normal University (No. 14ZB07).

REFERENCES

[1] X. H. Liu, *Resolution-Based Automated Reasoning*, Academic Press, China, 1994 (in Chinese).
[2] K. Y. Qin, Y. Xu and Z. M. Song, Lattice-valued propositional logic LP(X)(I), *Fuzzy Sys. Math.*, vol.11, no.4, pp.5-10, 1997.
[3] K. Y. Qin and Y. Xu, Lattice-valued propositional logic (II), *J. Southwest Jiaotong University*, vol.2, no.1, pp.22-27, 1994.
[4] J. P. Robinson, A machine-oriented logic based on the resolution principle, *J. ACM*, vol.12, pp.23-41, 1965.
[5] J. R. Slagle, Automatic theorem proving with renamable and semantic resolution, *J. ACM*, vol.14, no.4, pp.687-697, 1967.
[6] Y. Xu, Lattice implication algebras, *J. Southwest Jiaotong University*, vol.89, no.1, pp.20-27, 1993 (in Chinese).
[7] Y. Xu, D. Ruan, E. E. Kerre and J. Liu, α-resolution principle based on lattice-valued propositional logic LP(X), *Inform. Sci.*, vol.130, pp.195-223, 2000.
[8] Y. Xu, D. Ruan, E. E. Kerre and J. Liu, α-resolution principle based on lattice-valued first-order lattice-valued logic LF(X), *Inform. Sci.*, vol.132, pp.221-239, 2001.
[9] Y. Xu, D. Ruan, K. Y. Qin and J. Liu, *Lattice-Valued Logic*, Springer, 2003.
[10] Y. Xu, S. W. Chen, J. Liu and D. Ruan, Weak completeness of resolution in a linguistic truth-valued propositional logic, *Proc. of Theoretical Advances and Applications of Fuzzy Logic and Soft Computing*, Cancun, Mexico, pp.358-366, 2007.
[11] Y. Xu, J. Liu, D. Ruan and X. B. Li, Determination of α-resolution in lattice-valued first-order logic LF(X), *Inform. Sci.*, vol.181, pp.1836-1862, 2011.

[12] Y. Xu, J. Liu, X. Zhong and S. Chen, Multi-ary α-resolution principle for a lattice-valued logic, *IEEE Trans. Fuzzy Systems*, vol.26, no.5, pp.898-912, 2013.

[13] Y. Liu, Y. Xu and X. M. Zhong, Multi-ary α-semantic resolution automated reasoning based on lattice-valued first-order logic LF(X), *Journal of Intelligent and Fuzzy Systems*, vol.29, no.4, pp.1581-1593, 2015.

[14] Y. Liu, X. Y. Qin and Y. Xu, Interval-valued T-fuzzy filters and interval-valued T-fuzzy congruences on residuated lattices, *Journal of Intelligent and Fuzzy Systems*, vol.26, no.2, pp.2021-2033, 2014.

[15] Y. Liu, X. Y. Qin and Y. Xu, Determination of 3-ary α-resolution in lattice-valued propositional logic LP(X), *International Journal of Computational Intelligence Systems*, vol.6, no.5, pp.943-953, 2013.

[16] R. R. Yager, Inference in a multiple-valued logic system, *Int. J. Man Machine Stud.*, vol.23, pp.27-44, 1985.

[17] X. M. Zhong, Y. Xu, J. Liu, D. Ruan and S. W. Chen, General form of α-resolution based on linguistic truth-valued lattice-valued logic, *Soft Comput.*, vol.16, no.10, pp.1767-1781, 2012.

[18] X. M. Zhong, *On α-Quasi-Lock Semantic Resolution Automated Reasoning Based on Lattice-Valued Logic*, Ph.D. Thesis, Southwest Jiaotong University, 2012.

[19] J. F. Zhang and Y. Xu, Semantic resolution method on lattice-valued propositional logic LP(X), *Journal of Liaoning Technical University (Natural Science)*, vol.29, no.5, pp.767-770, 2010.

[20] J. F. Zhang, Y. Xu and X. X. He, Lattice-valued semantic resolution reasoning method, *Computer Science*, vol.38, no.9, pp.201-203,210, 2011.