Non-Clausal Multi-ary α-Generalized Resolution Calculus for a Finite Lattice-Valued Logic

Yang Xu[1], Jun Liu[2], Xingxing He[1]*, Xiaomei Zhong[1], and Shuwei Chen[1]*

[1]School of Mathematics, Southwest Jiaotong University
Chengdu 610031, P.R. China
{xuyang, x.he, zhongxm2013, swchen}@home.swjtu.edu.cn

[2] School of Computing, Ulster University
Northern Ireland, UK
j.liu@ulster.ac.uk

Abstract

Due to the need of the logical foundation for uncertain information processing, development of efficient automated reasoning system based on non-classical logics is always an active research area. The present paper focuses on the resolution-based automated reasoning theory in a many-valued logic with truth-values defined in a lattice-ordered many-valued algebraic structure - lattice implication algebras (LIA). Specifically, as a continuation and extension of the established work on binary resolution at a certain truth-value level α (called α-resolution), a non-clausal multi-ary α-generalized resolution calculus is introduced for a lattice-valued propositional logic LP(X) based on LIA, which is essentially a non-clausal generalized resolution avoiding reduction to normal clausal form. The new resolution calculus in LP(X) is then proved to be sound and complete. The concepts and theoretical results are further extended and established in the corresponding lattice-valued first-order logic LF(X) based on LIA.

Keywords: Automated reasoning; Resolution principle; Lattice-valued logic; Lattice implication algebra; Non-clausal multi-ary α-generalized resolution.

---

* The corresponding authors: x.he@home.swjtu.edu.cn; swchen@swjtu.edu.cn

1. Introduction

Automatic theorem proving is mechanization of mathematical inference by means of a logic system and inference rules, rather than computation over numbers. The resolution principle is one simple but efficient inference principle, and the resolution calculus is sound and complete in the Boolean logic system. Since its introduction by Robinson in 1965 [1], resolution-based automated reasoning has been extensively studied with the attempt to find natural and efficient proof systems to supporting a wide spectrum of computational tasks [2-5]. A number of important applications of resolution-based automated reasoning systems have been found in many areas such as artificial intelligence [6], logic programming [7, 8], problem solving [9], software model checking and testing [10], data structure verification [11], security protocols automated verification [12], question answering systems [13, 14], inconsistency checking for knowledge based system [15], and so on.

Many-valued logics become increasingly important in computer science and artificial intelligence aiming at establishing the logical foundation of uncertain information processing. Up to now, many researchers have made investigation on resolution-based automated reasoning in the framework of fuzzy logic and many-valued logics, and obtained some important results, refer to Section 4 – Related Work for some details.

To deal with sets of general formulas which can describe complex problems naturally, generalized resolution principle has been put forward for Boolean logic and many-valued logic as well, see [16-21], among others. In generalized resolution, the concept of literal is extended to generalized literal (which is composed by atomic formulas, 0, 1 and logical connectives), the resolvents can be obtained not only by the complementary literals from two conventional clauses, but also atomic formulas from any generalized clauses. It is more natural to retain the original forms in the statement of a theorem rather than transform them into several conjunctive normal forms (CNF). These transformations may produce high amount of new transformed formulas which may increase the complexity of automated reasoning process. In generalized resolution, there is no need to convert the formulas to the corresponding conjunctive normal forms. Meanwhile, the determination of whether two atoms are resolvable can be based on their syntactical form. Therefore, generalized resolution has been recognized as a more efficient proof method in implementation.

The present work focused specially on resolution-based automated reasoning in a many-valued logic L(X) with truth-values defined in a lattice-ordered many-valued algebraic structure - lattice implication algebras (LIA) [22, 23]. In order to handle more than two generalized clauses simultaneously, Xu *et al.* (2013) [24] extended the α-resolution principle in [25, 26] to multi-ary α-resolution principle in L(X), which can enhance resolution automated reasoning compared with the ones in classical logic and those proposed in some many-valued logics in terms of soundness and completeness, applicability, reasoning capability and reasoning efficiency. It has specially demonstrated clearly in [24] that binary resolution has limited reasoning capability and also reasoning efficiency especially in many-valued logic.

The above mentioned methods, however, can only deal with the generalized conjunctive normal form (GCNF) in L(X), but cannot handle other general forms of logical formulas. It means that any formula should be transformed into a GCNF in order to conduct the resolution deduction, but such transformation is also a complex process for pretreatment. Inspired from all the above ideas and motivation of generalized resolution in Boolean logic and multi-ary α-resolution in lattice-valued logic, the present paper aims to propose the *general generalized α-resolution principle* (the reason why it is called "general" and "generalized" will be clarified further in Section 3) in order to deal with complex formulas in finitely lattice-valued logic L(X). This paper is a continuation and extension of the work in [25-33], the binary α-resolution principle introduced in [25, 26] for L(X) is extended to multi-ary α-generalized resolution principle in different ways as follows: (1) the resolution is based on general generalized clauses which is constructed by the generalized literals and logical connectives "$\vee, \wedge, ', \rightarrow, \leftrightarrow$", instead of the generalized clause containing only "$', \rightarrow$" in [25, 26]. This, in essential, is a non-clausal resolution; (2) the set of the

generalized clauses, which is a complex logical formula, are not necessary to be transformed into the GCNF; (3) the above extended <u>binary</u> α-generalized resolution is further extended into <u>multi-ary</u> α-generalized resolution, i.e., extends the α-generalized resolution pair composed of two generalized literals to the α- generalized resolution group composed of multiple generalized literals based on the work in [24].

The remaining part of this paper is organized as follows. After a brief overview about lattice-valued logic based on LIA in Section 2, the definitions of the general form of α-generalized resolvent and α-generalized resolution deduction in LP(X), along with the discussion of the soundness and completeness of this ground case, are presented in Section 3. In Section 4, the soundness of the general form of α-generalized resolution deduction in LF(X) is given. In order to get the completeness of this general form of α-generalized resolution principle in LF(X), the α-unsatisfiability of the general form of logical formula is discussed. Finally, by Lifting Lemma and the completeness of the ground case, the completeness of the general case is obtained. Section 5 provides a review of some related work. The paper is concluded in Section 6.

2. Preliminaries

In this section, we only recall some elementary definitions, notations and properties needed in the subsequent discussions, more details about LIA, lattice-valued logic systems based on LIA and α-resolution principle introduced in those lattice-valued logic systems can be found in [22-26, 28].

2.1 Lattice implication algebra (LIA)

Definition 2.1 [22, 23] Let $(L, \vee, \wedge, O, I)$ be a bounded lattice with an order-reversing involution $'$, I and O the greatest and the smallest element of L respectively, and $\rightarrow: L \times L \dashrightarrow L$ be a mapping. $(L, \vee, \wedge, ', \rightarrow, O, I)$ is called a lattice implication algebra (LIA) 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) $(x \wedge y) \rightarrow z = (x \rightarrow z) \vee (y \rightarrow z)$.

Note that Boolean algebra and Łukasiewicz algebra, as two of the most popular logical algebras, are special cases of LIAs. In addition, LIAs form a proper class, and include no-chain algebra and no-Boolean algebra as well. The relationship between LIA with other logical algebraic structure is discussed in [24]. It shows that all the results obtained based on LIA or related logic can be applied into Boolean logic or Łukasiewicz logic at least, as well as other logical algebras. This reflects the fact that the investigation of resolution-based automated reasoning for lattice-valued logic in LIA is worthwhile in terms of generality and applicability.

2.2 Lattice-valued propositional logic LP(X)

Definition 2.2 [23] Let X be the set of propositional variables, $(L, \vee, \wedge, ', \rightarrow, O, I)$ be an LIA, $T = L \cup \{', \rightarrow\}$ be a type with ar$(') = 1$, ar$(\rightarrow) = 2$ and ar$(a) = 0$ for any $a \in L$. *The proposition logic algebra of the lattice-valued proposition calculus on the set X of propositional variables* (in short *lattice-valued proposition logic system*) is the free T algebra on X and denoted by LP(X).

Remark 2.1: Note that LP(X) includes the constant formulae $(a \in L)$, which has been one of the key differences from the most existing many-valued logic systems, because, with the constant formulae included and the implication connective defined differently from the Kleene implication, the syntax in LP(X) is essentially not equivalent to the one in the classical logic any more.

Definition 2.3 [23] *The set Γ of formula of* LP(X) is the least set Y satisfying the following conditions:
(1) $X \subseteq Y$;
(2) $L \subseteq Y$;
(3) if $p, q \in Y$, then $p', p \rightarrow q \in Y$.

Note that the set of all the L-type fuzzy sets on Γ is denoted by $\mathscr{F}_L(\Gamma)$. If $A \in \mathscr{F}_L(\Gamma)$, for any $G \in \Gamma$, $A(G) \in L$ represents the membership degree of G in A.

Definition 2.4 [23] A mapping γ: (LP(X), $'$, $\rightarrow$, O) $\rightarrow$

$(L, ', \rightarrow, O)$ is called a valuation of LP(X), if it is a T-homomorphism.

If γ is a valuation of LP(X), we have $\gamma(\alpha) = \alpha$ for any $\alpha \in L$. The special element O denotes *false* in (LP(X), $', \rightarrow, O)$.

Definition 2.5 [25] Let $G \in \Gamma$ and $\alpha \in L$. For any valuation γ of LP(X), if $\gamma(G) \leq \alpha$, we say *G is always less than or equal to α* (or G is α-*false*), denoted by $G \leq \alpha$.

2.3 Lattice-valued first-order logic LF(X)

Definition 2.6 [26] Suppose V and F are the set of variable symbols and that of functional symbols in LF(X), respectively, the set of terms of LF(X) is defined as the smallest set $\mathscr{T}$ satisfying the following conditions:

(1) $V \subseteq \mathscr{T}$;

(2) For any $n \in N$, if $f^{(n)} \in F$, then for any $t_1, \ldots, t_n \in \mathscr{T}, f^{(n)}(t_1, \ldots, t_n) \in \mathscr{T}$.

Remark 2.2: $f^{(0)}$ is specified as a constant symbol.

Definition 2.7 [26] Suppose P is a predicate symbol set in LF(X). The set of atoms of LF(X) is defined as the smallest set $\mathscr{A}_t$ satisfying the following condition:

For any $n \in N$, if $P^{(n)} \in P$, then $P^{(n)}(t_0, t_1, \ldots, t_n) \in \mathscr{A}_t$ for any $t_0, t_1, \ldots, t_n \in \mathscr{T}$.

Remark 2.3: $P^{(0)}$ is specified as a certain element in L.

Definition 2.8 [26] The set of formulas of LF(X) is defined as the smallest set $\mathscr{F}$ satisfying the following conditions:

(1) $\mathscr{A}_t \subset \mathscr{F}$; (2) If $p, q \in \mathscr{F}$, then $p \rightarrow q \in \mathscr{F}$;

(3) If $p \in \mathscr{F}$ and x is a free variable in p, then $(\forall x) p, (\exists x) p \in \mathscr{F}$.

Remark 2.4: Note that $p' = p \rightarrow O$, $p \vee q = (p \rightarrow q) \rightarrow q$, $p \wedge q = (p' \vee q')'$, $p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$. Therefore, if $p, q \in \mathscr{F}$, then $p', p \vee q, p \wedge q, p \leftrightarrow q \in \mathscr{F}$.

Definition 2.9 [26] Suppose $G \in \mathscr{F}$, F_G is the set of all functional symbols occurring in G, P_G is the set of all predicate symbols occurring in G, and $D (\neq \varnothing)$ is the domain of interpretations. An interpretation of G over D is a triple $I_D = \langle D, \mu_D, \nu_D \rangle$, where,

$$\mu_D : F_G \rightarrow U_D = \{ f_D^{(n)} : D^n \rightarrow D \mid n \in N \}$$

$$f^{(0)} \mapsto f_D^{(0)}, \quad f_D^{(0)}(D^0) = \{ f_D^{(0)} \} \subseteq D, \ D^{(0)} \text{ is a}$$

non-empty set

$$f^{(n)} \mapsto f_D^{(n)} \ (n \in N^+),$$

$$\nu_D : P_G \rightarrow V_D = \{ p_D^{(n)} : D^n \rightarrow L \mid n \in N \}$$

$$p^{(0)} \mapsto p_D^{(0)}, \quad p_D^{(0)}(D^0) = \{ p_D^{(0)} \} \subseteq L$$

$$p^{(n)} \mapsto p_D^{(n)} \ (n \in N^+).$$

3. Non-Clausal Multi-ary α-Generalized Resolution Principle for LP(X)

In this section, the outlined work in [29] will be extended and systematized, and further extended into the one for first-order logic LF(X) in Section 4.

In the following, the definition of literal is the same as that in Boolean logic, that is, a *literal* is either a propositional logic variable l or its negation $\sim l$.

Definition 3.1 [25] Let $G \in \Gamma$. G is called *an extremely simple form*, in short *ESF*, if $G^* (\in \Gamma)$ obtained by deleting any constant or literal or implication term occurring in G is not equivalent to G.

Definition 3.2 [25] Let $G \in \Gamma$. G is called *an indecomposable extremely simple form*, in short *IESF*, if the following two conditions hold:

(1) G is an *ESF* containing connectives $\rightarrow$ and $'$ at most;

(2) For any $H \in \Gamma$, if $H \in \overline{G}$ in $\overline{LP(X)}$, then H is an *ESF* containing connectives $\rightarrow$ and $'$ at most,

$$\overline{LP(X)} = (^{LP(X)}\!\!/_=, \vee, \wedge, ', \rightarrow, \overline{o}, \overline{I}) \text{ is the LIA,}$$

$^{LP(X)}\!\!/_= = \{ \overline{p} \mid p \in LP(X) \}, \quad \overline{p} = \{ q \mid q \in LP(X), q = p \},$

for any $\overline{p}, \overline{q} \in {}^{LP(X)}\!\!/_=$, $\overline{p} \vee \overline{q} = \overline{p \vee q}$, $\overline{p} \wedge \overline{q} = \overline{p \wedge q}$, $(\overline{p})' = \overline{p'}$, $\overline{p} \rightarrow \overline{q} = \overline{p \rightarrow q}$.

Definition 3.3 [25] All the constants, literals and *IESF*s in LP(X) are called *generalized literals* (*g-literals*). The constants, literals and *IESF*s in LP(X) are regarded as atomic formulas in LP(X).

Remark 3.1 In LP(X), a g-literal concept is normally with respect to a specific generalized clause. For example, let $C = (x \to y) \to z) \vee (z \to x)$, although $x \to y$ is a g-literal itself individually, but is not regarded as a g-literal in C, actually $(x \to y) \to z$ as an IESF is a g-literal in C.

Definition 3.4 [25] Let $G \in \Gamma$. G is called *a generalized clause* (*g-clause*), if G is a formula of the form below:

$$G = g_1 \vee \ldots \vee g_i \vee \ldots \vee g_n$$

where g_i ($i = 1, \ldots, n$) are g-literals in G. A conjunction of finite g-clauses is called *a generalized conjunctive normal form* (GCNF). In this paper, the empty clause is denoted as $\forall$.

For example, suppose x, y, z, s, t are propositional variables in LP(X), $\beta \in L$. Then $g_1 = x \to y$, $g_2 = y \to \beta$, $g_3 = (x \to z)'$, $g_4 = s$, and $g_5 = (x \to t') \to z$ are five g-literals; $C_1 = (x \to y) \vee (y \to \beta)$, $C_2 = (x \to z)' \vee s$, $C_3 = (y \to \beta) \vee (x \to z)' \vee ((x \to t') \to z)$ are three g-clauses; and $S = C_1 \wedge C_2 \wedge C_3$ is a GCNF.

Definition 3.5 [25] (α-Resolution). Let $\alpha \in L$, and G_1 and G_2 be two g-clauses in LP(X) of the forms below respectively:

$$G_1 = g_1 \vee \ldots \vee g_i \vee \ldots \vee g_m, \quad G_2 = h_1 \vee \ldots \vee h_j \vee \ldots \vee h_n.$$

If $g_i \wedge h_j \leq \alpha$, then

$$G = g_1 \vee \ldots \vee g_{i-1} \vee g_{i+1} \vee \ldots \vee g_m \vee h_1 \vee \ldots \vee h_{j-1} \vee h_{j+1}$$
$$\vee \ldots \vee h_n$$

is called an *α-resolvent of G_1 and G_2*, denoted by $G = R_\alpha(G_1, G_2)$, and g_i and h_j form an *α-resolution pair*, denoted by (g_i, h_j)-α. The generation of an α-resolvent from two clauses, called as *α-resolution*, is the sole rule of inference of the α-resolution principle, which is a generalization of *O-resolution* in the classical logic.

Definition 3.6 [25] In LP(X), suppose a GCNF $S = C_1 \wedge C_2 \wedge \ldots \wedge C_n$, $\alpha \in L$. $w = \{D_1, D_2, \ldots, D_m\}$ is *an α-resolution deduction from S to a g-clause D_m*, if for any $i \in \{1, 2, \ldots, m\}$

(1) $D_i \in \{C_1, C_2, \ldots, C_n\}$; or
(2) there exist j, $k < i$, such that $D_i = R_\alpha(D_j, D_k)$.

Remark 3.2 Specially, if there exists an α-resolution deduction from S to a clause which is α-false (also called *an α-false clause*, denoted by α-$\square$ to differentiate it from that in Boolean logic), then this α-resolution deduction w is called *an α-refutation*.

Different from some concepts in Boolean logic and also the *g-clause* in Definition 3.4 in LP(X), in the following, the key concept of this paper is introduced, that is, α-generalized resolution.

A concept of *g-clause* is recalled in Definition 3.4. It is called "generalized" due to the fact that it is based on the disjunction of *g-literals* (Definition 3.3), where g-literals could be constants, literals or *IESF*s in LP(X), and are regarded as atomic formulae. Here we consider more general cases, i.e., *general generalized clause*, which are actually a composite formula from the g-literals connected by logical connectives $\wedge$, or $\vee$, or $\to$, or $'$ or $\leftrightarrow$, that is, they are not limited to atomic formula, could be any single or composite formula, so it is call "general".

Definition 3.7 Let $g_1, \ldots, g_n$ be g-literals in LP(X). A lattice-valued propositional logical formula in LP(X) is called *a general generalized ground clause* (*general g^2-clause*), denoted by $\Phi(g_1, \ldots, g_n)$, if it is a composite formula from the g-literals $g_1 \ldots, g_n$ connected by logical connectives $\wedge$, or $\vee$, or $\to$, or $'$ or $\leftrightarrow$..

We use some examples to clarify the meaning of "general g^2-clause". For example, $\Phi_1 = (y \to z)' \wedge (w \to s)$, $\Phi_2 = (x \to y) \to z$, $\Phi_3 = x \to ((y \to a) \to t)$, $\Phi_4 = x \vee a$, $\Phi_5 = (y \to z) \leftrightarrow (x \to t')$, and $\Phi_6 = ((y \to z)' \wedge (w \to s)) \vee ((y \to z) \leftrightarrow (x \to t'))$, are all the general g^2-clauses, where x, y, z, w, s, t are propositional variables in LP(X), $a \in L$. Obviously, Φ_4 is a g-clause itself. Φ_6 is also a general g^2-clause although it is a combination of general g^2-clauses Φ_1 and Φ_5, which reflects the meaning of the term "general" here, i.e., any general composite formula by logical connectives $\wedge$, or $\vee$, or $\to$, or $'$ or $\leftrightarrow$ can be regarded as a general g^2-clause. From this point of view, Φ_2 and Φ_3 are also g-clauses as in Definition 3.4, but not for Φ_1, Φ_5 and Φ_6.

Based on Definition 3.7, it is easy to say that a composite formula from the g-literals connected by logical connectives ∧, or ∨, or →, or ′ or ↔ can be always transferred into a disjunctive form, and any lattice-valued propositional logical formula in LP(X) can be transferred into a conjunction of finite general g^2-clauses. These transformations are based on rules about logical equivalences [23]: the double negative law, the De Morgan's law and the distributive law between ∧ and ∨.

Here we still use the term "clause" to reflect the similar idea as the one using CNF in Boolean logic. As recalled in Definition 3.4, a conjunction of finite g-clauses is called *a generalized conjunctive normal form* (GCNF). Accordingly, a conjunction of finite general g^2-clauses is called *a general generalized ground conjunctive normal form* (G^3CNF). For example, $\Phi = ((x \to y) \to z) \wedge ((y \to z)' \vee (w \to s)) \wedge (x \to ((y \to z) \to t)) \wedge ((y \to z) \leftrightarrow (x \to t)')$ is a G^3CNF, i.e., a conjunction of 4 general g^2-clauses, where x, y, z, w, s, t are propositional variables in LP(X).

In essential, although we still use the term "clause" as the part of new concept, which actually is a kind of non-clausal formula compared with the one in Boolean logic.

Definition 3.8 A general g^2-clause G in LP(X) is called a *constant g^2-clause* if all the g-literals in G are all constants. Particularly, if $\gamma(G) = \alpha$ for any valuation γ of LP(X), then G is called an *α-constant g^2-clause*.

Definition 3.9 Let Φ be a general g^2-clause in LP(X). A g-literal g of Φ is called *a local extremely complex form*, if

(1) g can't be expanded to a more complex g-literal in Φ by adding → and ′; or

(2) If $g = g_1 \leftrightarrow g_2$, g_1 and g_2 are g-literals in LP(X), then g is a local extremely complex form as a whole.

Example 3.1 Let Φ be a general g^2-clause in LP(X), $\Phi = ((x \to y) \to z) \wedge ((y \to z)' \vee (w \to s)) \wedge (x \to ((y \to z) \to t)) \wedge ((y \to z) \leftrightarrow (x \to t)')$, where x, y, z, w, s, t are propositional variables in LP(X). For $(x \to y) \to z$, the local extremely complex form should not be generalized literals x, y, z, $x \to y$, but $(x \to y) \to z$ itself. Hence the local extremely complex forms in Φ are $(x \to y) \to z$, $(y \to z)'$, $w \to s$, $x \to ((y \to z) \to t)$, and $(y \to z) \leftrightarrow (x \to t)'$.

Remark 3.3 In the following, all the g-literals discussed are local extremely complex forms.

In the following discussion, we extend α-resolution for LP(X) introduced in [25] to non-clausal multi-ary α-generalized resolution in LP(X), i.e., (1) extend from α-resolution to α-generalized resolution; (2) extend from binary resolution to multi-ary resolution, i.e., the resolved generalized literals extended from two g-literals to the finite g-literals for batch processing clauses, so it can be regarded as a kind of group resolution; and also (3) extend from multi-ary resolution to non-clausal multi-ary resolution. These extensions can improve the efficiency and applicability of α-resolution, and feasible implementation algorithms can be much easier to be established based on it.

Accordingly, the concept of non-clausal multi-ary α-generalized resolution principle is firstly introduced in LP(X), along with the corresponding non-clausal multi-ary α-generalized resolution deduction; its soundness and completeness are also proved afterwards.

Definition 3.10 (Non-clausal n-ary α-generalized resolution) Let Φ_1, Φ_2,..., Φ_n be general g^2-clauses in LP(X), H_i be the set of g-literals in Φ_i, $\alpha \in L$. If there exist general literals $g_i \in H_i$ (i=1, 2,..., n), such that $\wedge_{i=1}^{n} g_i \leq \alpha$ (i.e., α false), then $G = \vee_{i=1}^{n} \Phi_i (g_i = \alpha)$ is called *a non-clausal n-ary α-generalized resolvent* of Φ_1, Φ_2, ..., Φ_n, denoted by $G = R_{p(\text{N-}n\text{-}\alpha)\text{-}g} (\Phi_1 (g_1), \Phi_2 (g_2), ..., \Phi_n (g_n))$, here "$p$" means "propositional logic" and "(N-n-α)-g" means "non-clausal n-ary α-generalized" in "$R_{p(\text{N-}n\text{-}\alpha)\text{-}g}$".

Theorem 3.1 Let Φ_1, Φ_2,..., Φ_n be general g^2-clauses in LP(X), H_i is the set of g-literals in Φ_i ($i = 1, 2, ..., n$), $\alpha \in L$. If there exist g-literals $g_i \in H_i$ ($i = 1, 2, ..., n$), such that $\wedge_{i=1}^{n} g_i \leq \alpha$, then

$$\wedge_{i=1}^{n} \Phi_i \leq R_{p(\text{N-}n\text{-}\alpha)\text{-}g} (\Phi_1 (g_1), \Phi_2 (g_2), ..., \Phi_n (g_n)).$$

Proof. It follows from the definition of the general g^2-clauses and the logical equivalences in LP(X) that Φ_i ($i = 1, 2, ..., n$) could be converted to the corresponding generalized conjunction normal forms (GCNF), i.e., $\Phi_i = \wedge_{j=1}^{n_i} G_{ij}$, where G_{ij} is a g-clause. Hence, Φ_i can be

rewritten as

$$\Phi_i = \wedge_{j=1}^{n_i} G_{ij} = \wedge_{j=1}^{m_i} (g_i \vee \bigvee_{g \in H_{ij}-\{g_i\}} g) \wedge G^0{}_i, \quad i = 1, 2, \ldots, n,$$

where m_i is the g-clause number which the g-clause G_{ij} in $\{ G_{ij}/j=1,\ldots, n_i \}$ includes g_i, H_{ij} is the g-literals set of G_{ij} including g_i, and $G^0{}_i$ is the conjunction of all G_{ij} ($j = 1, 2,\ldots, n_i$) which doesn't include g_i. Note that

$$\wedge_{i=1}^{n} [\ \wedge_{j=1}^{m_i} (g_i \vee \bigvee_{g \in H_{ij}-\{g_i\}} g\ )] = \wedge_{i=1}^{n} g_i \vee$$

$$\bigvee_{y \in \prod_{i=1}^{n}\{g_i,(\wedge_{j=1}^{m_i} \vee_{g \in H_{ij}-\{g_i\}} g)\}-\{(g_1,g_2,\ldots,g_n)\}} (\wedge_{l=1, z_l \in y}^{n} z_l)$$

$$\leq \wedge_{i=1}^{n} g_i \vee \vee_{i=1}^{n}(\wedge_{j=1}^{m_i} \vee_{g \in H_{ij}-\{g_i\}} g) \leq \alpha \vee$$

$$\vee_{i=1}^{n}(\wedge_{j=1}^{m_i} \vee_{g \in H_{ij}-\{g_i\}} g), \text{ and}$$

$$\wedge_{i=1}^{n} \Phi_i = \wedge_{i=1}^{n} [\ \wedge_{j=1}^{m_i} (g_i \vee \bigvee_{g \in H_{ij}-\{g_i\}} g\ ) \wedge$$

$$G^0{}_i] = \wedge_{i=1}^{n} [\ \wedge_{j=1}^{m_i} (g_i \vee \bigvee_{g \in H_{ij}-\{g_i\}} g)] \wedge \wedge_{i=1}^{n} G^0{}_i,$$

$$\leq [\alpha \vee \vee_{i=1}^{n}(\wedge_{j=1}^{m_i} \vee_{g \in H_{ij}-\{g_i\}} g)] \wedge$$

$$\wedge_{i=1}^{n} G^0{}_i = \vee_{i=1}^{n} \wedge_{j=1}^{m_i} (\alpha \vee \vee_{g \in H_{ij}-\{g_i\}} g) \wedge \wedge_{i=1}^{n} G^0{}_i$$

$$\leq \vee_{i=1}^{n}[\wedge_{j=1}^{m_i}(\alpha \vee \vee_{g \in H_{ij}-\{g_i\}} g) \wedge G_i^0]$$

$$= \vee_{i=1}^{n} \Phi_i(g_i = \alpha)$$

$$= R_{p(N-n-\alpha)-g}(\Phi_1(g_1), \Phi_2(g_2), \ldots, \Phi_n(g_n)).$$

Therefore, $\wedge_{i=1}^{n} \Phi_i \leq R_{p(N-n-\alpha)-g}(\Phi_1(g_1), \ldots, \Phi_n(g_n))$.

Remark 3.4 (1) In the proof of Theorem 3.1, the general g^2-clauses should be converted to the corresponding generalized conjunctive normal forms. However, in the course of α-generalized resolution, the general g^2-clauses need not to be converted.

(2) If a g-literal which includes many implication connectives is not considered as a local extremely complex form, then Theorem 3.1 may not hold. An example is shown as follows.

Example 3.2 [23] Let $L_6 = \{ O, a, b, c, d, I \}$, $O' = I$,

7

$a' = c$, $b' = d$, $c' = a$, $d' = b$, $I' = O$, the Hasse diagram of L_6 be defined as Figure 3.1 and its implication operator be defined as Table 3.1. Then $(L_6, \vee, \wedge, ', \to, O, I)$ is an LIA.

Table 3.1 Implication Operator of L_6

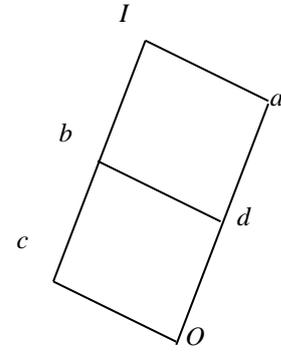| $\to$	O	a	b	c	d	I
O	I	I	I	I	I	I
a	c	I	b	c	b	I
b	d	a	I	b	a	I
c	a	a	I	I	a	I
d	b	I	I	b	I	I
I	O	a	b	c	d	I

Figure 3.1 Hasse Diagram of L_6

Example 3.3 Let Φ_1 and Φ_2 be two general g^2-clauses in $L_6P(X)$, $\Phi_1 = (b \to x)' \to d$, $\Phi_2 = (d \to x)' \to b$, where x is a propositional variable, $b, d \in L_6$, and let the resolution level $\alpha = a \in L_6$ (see Example 3.2). So $\Phi_1 \wedge \Phi_2 = ((b \to x)' \to d) \wedge ((d \to x)' \to b) \geq (b \to d) \wedge (d \to b) = a$. On the other hand, since $(b \to x)' \wedge (d \to x)' \leq b \wedge d = d < a$, if $g_1 = (b \to x)'$, $g_2 = (d \to x)'$, then $R_{p(N-2-\alpha)-g}(\Phi_1(g_1), \Phi_2(g_2)) = (a \to d) \vee (a \to b) = b \vee b = b$. However, $\Phi_1 \wedge \Phi_2 \not\leq R_{p(N-2-\alpha)-g}(\Phi_1(g_1), \Phi_2(g_2))$ for $a \not\leq b$.

Definition 3.11 A lattice-valued propositional logical formula S in LP(X) is called *a general g^2-clause set* or *general generalized ground conjunctive normal form (G^3CNF)* if S is a formula of the form $S = \Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$, where Φ_i ($i = 1, 2, \ldots, n$) are general g^2-clauses.

Remark 3.5 The formula S can also be denoted as

$S = \{ \Phi_1, \Phi_2, \ldots, \Phi_n \}$ for short.

Definition 3.12 Suppose S is a G^3CNF in LP(X), $\alpha \in L$. Then the sequence $D_1, D_2, \ldots, D_m$ is called a non-clausal multi-ary α-generalized resolution deduction from S to a general g^2-clause D_m, if

(1) $D_i \in S$ ($i = 1, 2, \ldots, m$); or

(2) There exist $r_1, r_2, \ldots, r_{k_i} < i$, such that

$R_{p(\text{N-}k_i\text{-}\alpha)\text{-g}} (Dr_1, Dr_2, \ldots, D r_{k_i}) = D_i.$

If there exists a non-clausal multi-ary α-generalized resolution deduction w from S to the α-false clause (denoted by α-□ similar to the one in Remark 3.2), then w is called *a non-clausal* multi-ary *α-refutation.*

Theorem 3.2 (Soundness) Let S be a set of general g^2-clauses in LP(X), $\alpha \in L$. the sequence D_1, $D_2, \ldots, D_m$ be a non-clausal multi-ary α-generalized resolution deduction from S to a general g^2-clause D_m. If $D_m = \alpha$-□, then $S \leq \alpha$.

Proof. From Definition 3.10 and Theorem 3.1, it is easy to obtain that $S = S \wedge D_1 \wedge D_2 \wedge \ldots \wedge D_m \leq D_m = \alpha$-□.

Theorem 3.3 (Completeness) Suppose S is the set of general g^2-clauses $\Phi_1, \Phi_2, \ldots, \Phi_n$ in LP(X). If $S \leq \alpha$, then there exists a non-clausal multi-ary α-generalized resolution deduction from S to α-□.

Proof. Suppose H_i is the set of g-literals in Φ_i ($i=1, 2, \ldots, n$). Let $H = U_{i=1}^{n} H_i$ and $|H|$ be the number of elements in H. We will prove Theorem 3.3 by induction on $|H|$.

If $|H| = 1$, then there exists a g-literal g, such that $S = g \leq \alpha$. Hence $g(g = \alpha) = \alpha$, i.e., S can be α-generalized resolved into α-□.

If $|H| = 2$, then $H = \{g_1, g_2\}$. Hence if $S = g_1 \wedge g_2$, we can obtain $g_1(g_1 = \alpha) \vee g_2(g_2 = \alpha) = \alpha$, i.e., S can be α-generalized resolved into α-$\forall$. If $S = g_1 \vee g_2$, we can obtain the same conclusion.

Suppose that Theorem 3.3 is true for $|H| < n$ ($n \geq 3$), we prove it also holds for $|H| = n$.

Let $S = \Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$. We consider two cases as follows.

Case 1. If Φ_i is the conjunction of the g-literals for $i =1, 2, \ldots, n$, i.e., $\Phi_i = \wedge_{j=1}^{n_i} g_{ij}$ for $i =1, 2, \ldots, n$, then $S = \wedge_{i=1}^{n} \wedge_{j=1}^{n_i} g_{ij}$. If $S \leq \alpha$, then $R_{p(\text{N-}n\text{-}\alpha)\text{-g}} (\Phi_1, \Phi_2, \ldots, \Phi_n)$

$= R_{p(\text{N-}\prod_{i=1}^{n} n_i \text{ -}\alpha)\text{-g}} (g_{11}, g_{12}, \ldots, g_{1n_1}, g_{21}, g_{22}, \ldots, g_{2n_2}, \ldots, g_{n1}, g_{n2}, \ldots, g_{nn_n})$

$= \wedge_{i=1}^{n} \wedge_{j=1}^{n_i} g_{ij}(g_{ij} = \alpha) = \alpha$. In this case, the conclusion holds.

Case 2. If there exists a general g^2-clause Φ_{i_0}, such that $\Phi_{i_0} = \Phi_{i_0}^{*} \vee g_{i_0}$, where g_{i_0} is a g-literal in Φ_{i_0}, $\Phi_{i_0}^{*}$ includes g-literal. Without loss of generality, suppose $i_0 = 1$, then $S = (\Phi_1^{*} \vee g_1) \wedge \Phi_2 \wedge \ldots \wedge \Phi_n = (\Phi_1^{*} \wedge \Phi_2 \wedge \ldots \wedge \Phi_n) \vee (g_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n)$. Let $S_1 = \Phi_1^{*} \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$, $S_2 = g_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$. Then $S_1 \leq \alpha$, $S_2 \leq \alpha$, $| H_{S_1} | < n$, and $| H_{S_2} | < n$, where H_{S_k} is the set of g-literals in S_k, $k=1,2$. By induction hypothesis, S_1 and S_2 respectively have non-clausal multi-ary α-generalized resolution deduction sequences as follows:

$D_{11}', D_{12}', \ldots, D_{1m_1}'$, where D_{1m_1}' is an α-constant g^2-clause;

$D_{21}', D_{22}', \ldots, D_{2m_2}'$, where D_{2m_2}' is an α-constant g^2-clause.

Now we renew Φ_1^{*} to Φ_1 in $D_{11}', D_{12}', \ldots, D_{1m_1}'$, and obtain the α-generalized resolution deduction

8

sequence from S to D_{1m_1} as follows.

D_{11}, D_{12}, …, D_{1m_1}, where (1) D_{1m_1} is an α-constant g^2-clause; or (2) D_{1m_1} is $\alpha \vee g_1$.

If D_{1m_1} is $\alpha \vee g_1$, for the first g-literal g_1 in S_2, we renew it to $\alpha \vee g_1$ in D_{21}', D_{22}', …, D_{2m_2}', and obtain the α-generalized resolution deduction sequence from $(\alpha \vee g_1) \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$ to D_{2m_2} as follows: D_{21}, D_{22}, …, D_{2m_2}, where D_{2m_2} is also an α-constant g^2-clause.

If it is Case (1), the conclusion holds.

If it is Case (2), then the α-generalized resolution deduction sequence

$D_{11}, D_{12}, \ldots, D_{2m_2}$ $(=\alpha \vee g_1), D_{21}, D_{22}, \ldots, D_{2m_2}$

is the α-generalized resolution deduction from S to α-constant g^2-clause.

This completes the proof.

The following two examples provide some illustration of completeness of non-clausal multi-ary α-generalized resolution deduction in LP(X).

Example 3.4 Let $L_9 = \{a_i \mid 1 \le i \le 9\}$ be a Łukasiewicz implication algebra (refer to Example 2.3, here n=9), x, y, z, u, v, w propositional variables in $L_9P(X)$, $S = \{x, (x \to a_2) \vee (z \to v)' \vee (u \leftrightarrow w), (y \to x)' \vee (u \leftrightarrow w)', z \to a_3, a_5 \to v\}$. We take the resolution level $\alpha = a_5$, then $S \le a_5$. By the completeness of non-clausal multi-ary α-generalized resolution in LP(X) (i.e., Theorem 3.3), there exists a non-clausal multi-ary a_5-generalized resolution refutation of S, for example, the one as follows:

(1) x
(2) $(x \to a_2) \vee (z \to v)' \vee (u \leftrightarrow w)$
(3) $(y \to x)' \vee (u \leftrightarrow w)'$
(4) $z \to a_3$

(5) $a_5 \to v$
----
(6) $a_5 \vee (z \to v)' \vee (u \leftrightarrow w)$ by (1), (2)
(7) $a_5 \vee (u \leftrightarrow w)'$ by (1), (3)
----
(8) $a_5 \vee (u \leftrightarrow w)$ by (4), (5), (6)
----
(9) a_5-□   by (7), (8)

Moreover, if we judge the α-unsatisfiability of S by α-resolution, we firstly transform S to its generalized conjunctive normal form $S_1 = \{x, (x \to a_2) \vee (z \to v)' \vee (u \to w), (x \to a_2) \vee (z \to v)' \vee (w \to u), (y \to x)' \vee (u \to w)' \vee (w \to u)', z \to a_3, a_5 \to v\}$. We cannot get a binary α-resolution refutation of S_1, but can get an alternative multi-ary α-resolution refutation of S_1 using the multi-ary α-resolution principle introduced in [Xu *et al.* 2013], however which apparently needs more steps or is relatively more complex, i.e.,

(1) x
(2) $(x \to a_2) \vee (z \to v)' \vee (u \to w)$
(3) $(x \to a_2) \vee (z \to v)' \vee (w \to u)$
(4) $(y \to x)' \vee (u \to w)' \vee (w \to u)'$
(5) $z \to a_3$
(6) $a_5 \to v$
----
(7) $a_5 \vee (z \to v)' \vee (u \to w)$ by (1), (2)
(8) $a_5 \vee (z \to v)' \vee (w \to u)$ by (1), (3)
(9) $a_5 \vee (u \to w)' \vee (w \to u)'$ by (1), (4)
----
(10) $a_5 \vee (u \to w)$ by (5), (6), (7)
(11) $a_5 \vee (w \to u)$ by (5), (6), (8)
----
(12) $a_5 \vee (w \to u)'$ by (9), (10)
(13) a_5-□   by (11), (12)

Example 3.5 Let $(L_6, \vee, \wedge, ', \to, O, I)$ be an LIA as defined in Example 3.2, x, z, s, t propositional variables in $L_6P(X)$, $S = \{x, (x \to a), (x \to b) \vee (t \to d), (s \leftrightarrow t) \vee (z \to t), (s \leftrightarrow t)', (z \to c)'\}$. We take the resolution level $\alpha = d$, then $S \le d$. By the completeness of non-clausal multi-ary α-generalized resolution in LP(X), there exists a non-clausal multi-ary d-generalized resolution refutation of S, for example, the one as follows:

(1) x

(2) $x \to a$

(3) $(x \to b) \vee (t \to d)$

(4) $(s \leftrightarrow t) \vee (z \to t)$

(5) $(s \leftrightarrow t)'$

(6) $(z \to c)'$

----

(7) $d \vee (t \to d)$ by (1), (2), (3)

(8) $d \vee (z \to t)$ by (4), (5)

----

(9) d-□   by (6), (7), (8)

Similarly, if we judge the α-unsatisfiability of S by α-resolution principle, we firstly transform S to its generalized conjunctive normal form $S_1 = \{x, (x \to a), (x \to b) \vee (t \to d), (s \to t) \vee (z \to t), (t \to s) \vee (z \to t), (s \to t)' \vee (t \to s)', (z \to c)'\}$. We cannot get a binary α-resolution refutation of S_1, but can get a relatively complex multi-ary α-resolution refutation of S_1 [Xu *et al.* 2013], i.e.,

(1) x

(2) $(x \to a)$

(3) $(x \to b) \vee (t \to d)$

(4) $(s \to t) \vee (z \to t)$

(5) $(t \to s) \vee (z \to t)$

(6) $(s \to t)' \vee (t \to s)'$

(7) $(z \to c)'$

----

(8) $d \vee (t \to d)$ by (1), (2), (3)

----

(9) $d \vee (s \to t)$ by (4), (7), (8)

(10) $d \vee (t \to s)$ by (5), (7), (8)

----

(11) $d \vee (t \to s)'$ by (6), (9)

----

(12) d-□   by (10), (11)

From the above two examples show that there is no need to transform those complex formula into the generalized conjunctive normal form (actually it is not straightforward for the transformation), so that means "non-clausal" resolution. Theorem 3.2 and Theorem 3.3 show that the non-clausal multi-ary α-generalized resolution deduction in LP(X) is sound and complete, along with Examples 3.4 and 3.5 to illustrate the distinct advantages of non-clausal multi-ary α-generalized resolution in terms of reasoning capability and efficiency. Furthermore, for an α-unsatisfiable set of general g²-clauses in LP(X), if we convert it into the corresponding generalized conjunctive normal form, and judge it by the α-resolution principle, then it may not lead to a binary α-resolution refutation partially because its restriction of the number of α-resloved literals is 2. Although sometimes it may lead to a multi-ary α-resolution refutation using the method in [24], it is more complex than the non-clausal multi-ary α-resolution refutation introduced in this paper.

4. Non-Clausal Multi-ary α-Generalized Resolution Principle for LF(X)

In this section, non-clausal multi-ary α-generalized resolution principle is extended from LP(X) to LF(X), as extended and systematized version of [30]. The α-unsatisfiability for a general form of the logical formula, i.e., general g-clause in LF(X) is discussed firstly. Then, Lifting Lemma of the non-clausal multi-ary α-generalized resolution principle for LF(X) is established, and the soundness of this general case is shown. Finally, by the completeness of the ground case, we lift it to the general case in LF(X), i.e., the completeness for this general form of non-clausal multi-ary α-generalized resolution deduction for LF(X).

4.1 α- unsatisfiability for a general form of the logical formula in LF(X)

Definition 4.1 [26] A formula G in lattice-valued first-order logic LF(X) is *a generalized-literal*, if it satisfies the following conditions:

(1) G is a literal; or

(2) G is constructed only by some literals and some implication connectives with the condition that G cannot be represented by connectives "$\vee$" or "$\wedge$" and G cannot be decomposed into a simpler form (G is called *an indecomposable implication form*).

The disjunction of a finite number of generalized-literals is a generalized-clause. The conjunction of a finite number of generalized-clauses is *a generalized-conjunctive normal form*.

Definition 4.2 [26] Let $G \in \mathscr{F}$, $\alpha \in L$. G is said to be

α-*false*, if $v_D(G) \leq \alpha$ for any interpretation $I_D = \langle D, \mu_D, v_D \rangle$ of G.

Definition 4.3 [26] Suppose G is a formula of the form $Q_1x_1...Q_nx_nG^*$, where $Q_1,..., Q_n$ are the quantifiers, i.e., $\forall$ or $\exists$, and G^* is a formula without any quantifier. Then G is said to be *a generalized-prenex conjunctive normal form*, if G^* is a generalized-conjunctive normal form.

Definition 4.4 [26] Suppose a formula $G = Q_1x_1...Q_nx_nM$ is a generalized-prenex conjunctive normal form. The formula G^* obtained by the following steps is called *a generalized-Skolem normal form of G*:

(1) If Q_r is an existential quantifier and without any universal quantifier occurring ahead it in the prefix $Q_1,..., Q_n$ (from left to right), we choose a new constant c different from other constants occurring in M, replace all x_r occurring in M by c, and then delete Q_r from the prefix $Q_1,..., Q_n$.

(2) If Q_r is an existential quantifier and $Q_{k_1},..., Q_{k_m}$ are all the universal quantifiers occurring ahead Q_r ($m \geq 1$, $1 \leq k_1 < ... < k_m < r$), we choose a new m-ary function symbol f_m^G different from all other function symbols occurring in M, replace all x_r in M by $f_m^G(x_{k_1},..., x_{k_m})$ and then delete Q_r from the prefix $Q_1,..., Q_n$.

(3) Repeating (1) and (2) until there is no existential quantifier occurring in the prefix.

Definition 4.5 Let $g_1, g_2, ..., g_n$ be g-literals in LF(X). A logical formula F in LF(X) is called a *general g-clause* if these g-literals are connected by logical connectives $\wedge$, $\vee$, $\rightarrow$, $'$ and $\leftrightarrow$, denoted by $\Phi(g_1, g_2,..., g_n)$.

Definition 4.6 A general g-clause in LF(X) is called a *constant g-clause* if it contains only constants. Particularly, for a constant g-clause G, if $v_D(G) = \alpha$ for any interpretation $I_D = <D, \mu_D, v_D>$, it follows that then this constant g-clause G is called an α-constant g-clause.

The conjunction of finite general g-clauses is a general g-conjunctive normal form. Similar to α-resolution principle in LF(X) [26], we can formally give definitions of general g-prenex normal form, general g-prenex conjunction normal form, general

g-Skolem normal form, etc. Also, the definitions such as substitution, the most general unifier, ground substitution, instance, ground instance occurring in the following are the same as those in Boolean logic.

Definition 4.7 Let Φ be a general g-clause in LF(X). A g-literal g of Φ is called *a local extremely complex form*, if

(1) g can't be expanded to a more complex g-literal in Φ by adding $\rightarrow$ and $'$; or

(2) If $g = g_1 \leftrightarrow g_2$, g_1 and g_2 are g-literals in LF(X), then g is a local extremely complex form as a whole.

Remark 4.1 In the following, all the g-literals discussed are local extremely complex forms.

Here, the g-clause is extended to the general g-clause, i.e., the general g-clause may not be the disjunction of the g-literals; it seems that it's too complex to explore the α-unsatisfiability of the logical formula in LF(X). Fortunately, note that every variable in the general g-Skolem normal form is universally quantified, and many properties of formulas in LF(X) do not rely on the structure of the g-clauses, many conclusions in the g-Skolem normal form still hold for general form as far as α-unsatisfiability is concerned. We only state main results as follows.

Theorem 4.1 Let G and G^* be two logical formulas in LF(X), and G^* a general g-Skolem normal form of G, $\alpha \in$ L. Then G is α-false if and only if G^* is α-false.

Proof. It is obvious from Theorem 2.1 in [26].

Theorem 4.2 Suppose G^* is a general g-Skolem normal form of a formula G in LF(X). If an interpretation $I_D = <D, \mu_D, v_D>$ α-satisfies G^*, then the H-interpretation $I_H = <H, \mu_H, v_H>$ of G^* corresponding to I_D also α-satisfies G^*.

Proof. It is obvious from Theorem 3.1 in [26].

Theorem 4.3 Suppose G^* is a general g-Skolem normal form of a formula G in LF(X). Then G^* is α-false if and only if $v_H(G^*) \not\geq \alpha$ holds for all H-interpretations $I_H = <H, \mu_H, v_H>$ of G^*.

Proof. It is obvious from Theorem 3.2 in [26].

Theorem 4.4 Suppose G^* is a general g-Skolem

normal form of a formula G in LF(X), $|L|<+\infty$. Then G^* is α-false if and only if there exists $K\in N$ such that $\nu_H(G^*) \not\geq \alpha$ holds for every adjoint ν_H of every element in L^K.

Proof. It is obvious from Theorem 3.3 in [26].

Theorem 4.5 Suppose G^* is a general g-Skolem normal form of a formula G in LF(X), $|L|<+\infty$. Then G^* is α-false if and only if there exists a finite ground instance set G^{*0} of G^* such that G_c^{*0} is α-false, where G_c^{*0} is the conjunction of all ground instances of G^{*0}.

Proof. It is obvious from Theorem 3.4 in [26].

From Theorem 4.5, in order to judge the α-unsatisfiability of a formula in LF(X), when $|L|<+\infty$, we only need to find a finite and ground instance set of this formula, and validate its α-unsatisfiability. Hence it is relatively achievable.

4.2 Soundness and completeness for the general form of non-clausal multi-ary α-generalized resolution deduction in LF(X)

This section provides the soundness and completeness for the general form of the non-clausal multi-ary α-generalized resolution deduction in LF(X).

Definition 4.8 Let Φ be a general g-clause in LF(X). If there exists a most general unifier σ of g-literals g_1, g_2, ..., g_m in Φ, then Φ^σ is called a *factor* of Φ.

Definition 4.9 (A general form of non-clausal multi-ary α-generalized resolution) Let Φ_1, Φ_2, ..., Φ_n be general g-clauses in LF(X), $\Phi_1^{\sigma 1}$ be a factor of Φ_1 for g-literals g_{11}, g_{12}, ..., g_{1r_1}, $\Phi_2^{\sigma 2}$ be a factor of Φ_2 for g-literals g_{21}, g_{22}, ..., g_{2r_2}, ..., and $\Phi_n^{\sigma n}$ be a factor of Φ_n for g-literals g_{n1}, g_{n2}, ..., g_{nr_n}, $\alpha\in L$. If $\wedge_{i=1}^{n} g_{i1}^{\sigma_i} \leq \alpha$ (i.e., α false), then $G = \vee_{i=1}^{n} \Phi_i^{\sigma_i} (g_{i1}^{\sigma_i} = \alpha)$ is called a *non-clausal n-ary α-generalized resolvent* of Φ_1, Φ_2, ..., Φ_n, denoted by $G = R_{f(N\text{-}n\text{-}\alpha)\text{-}g} (\Phi_1, \Phi_2, ..., \Phi_n)$, here "$f$" means "first-order logic".

Definition 4.10 Suppose S is a general g-clause set in LF(X), $\alpha \in L$, the sequence D_1, D_2, ..., D_m is called a *non-clausal multi-ary α-generalized resolution deduction* from S to general g-clause D_m, if

(1) $D_i \in S$ ($i = 1, 2, ..., m$); or

(2) There exist r_1, r_2, ..., $r_{k_i} < i$, such that $R_{f(N\text{-}k_i\text{-}\alpha)\text{-}g} (Dr_1, Dr_2, ..., D r_{k_i}) = D_i$.

The usual method for proving the completeness of a version of resolution has two steps. Firstly, one proves the ground case in propositional logic, in which no variables occur. Then one lifts it to the general case in first-order logic.

In the Proof of Theorem 5.3 (Lifting Lemma) for generalized clause in LF(X) [24], the disjunction of g-literals was not used, so Theorem 4.6 also holds.

Theorem 4.6 (Lifting Lemma) If Φ^0_1, Φ^0_2,..., Φ^0_n are instances of general g-clauses Φ_1, Φ_2,..., Φ_n in LF(X), respectively, P^0 is a non-clausal multi-ary α-generalized resolvent of Φ^0_1, Φ^0_2,..., Φ^0_n, then there exists a non-clausal multi-ary α-generalized resolvent P of Φ_1, Φ_2,..., Φ_n such that P^0 is an instance of P.

Theorem 4.7 Let Φ_1, Φ_2,..., Φ_n be general g-clauses in LF(X), $\Phi_1^{\sigma_1}$ be a factor of Φ_1 for g-literals g_{11}, g_{12}, ..., g_{1r_1}, $\Phi_2^{\sigma_2}$ be a factor of Φ_2 for g-literals g_{21}, g_{22}, ..., g_{2r_2}, ..., and $\Phi_n^{\sigma_n}$ be a factor of Φ_n for g-literals g_{n1}, g_{n2}, ..., g_{nr_n}, $\alpha\in L$. If $\wedge_{i=1}^{n} g_{i1}^{\sigma_i} \leq \alpha$ (i.e., α false), then

$$\wedge_{i=1}^{n} \Phi_i \leq \vee_{i=1}^{n} \Phi_i^{\sigma_i} (g_{i1}^{\sigma_i} = \alpha).$$

Proof. Similar to the proof of Theorem 3.1, for most general unifiers $\sigma_i (i = 1, 2, ..., n)$, if $\wedge_{i=1}^{n} g_{i1}^{\sigma_i} \leq \alpha$, then $\wedge_{i=1}^{n} \Phi_i^{\sigma_i} \leq \vee_{i=1}^{n} \Phi_i^{\sigma i} (g^{\sigma i}_{i1} = \alpha)$. Hence, it follows from $\wedge_{i=1}^{n} \Phi_i \leq \wedge_{i=1}^{n} \Phi_i^{\sigma_i}$ that $\wedge_{i=1}^{n} \Phi_i \leq \vee_{i=1}^{n} \Phi_i^{\sigma_i} (g_{i1}^{\sigma_i} = \alpha)$.

Theorem 4.8 (Soundness) Let S be a general g-clause set in LF(X), $\alpha \in L$. The sequence Φ_1, Φ_2, ..., Φ_m be a non-clausal multi-ary α-generalized resolution deduction from S to general g-clause Φ_m. If $\Phi_m = \alpha\text{-}\forall$, then $S \leq \alpha$, i.e., S is α false.

Proof. According to Theorem 4.7, similar to the proof of Theorem 3.2, we can easily get the conclusion.

Theorem 4.9 (Completeness) Suppose S is the set of general g-clauses Φ_1, Φ_2, ..., Φ_n in LF(X), $|L|<+\infty$. If $S \leq \alpha$ (i.e., S is α false), then there exists a non-clausal multi-ary α-generalized resolution deduction from S to $\alpha\text{-}\square$.

Proof. By Theorem 4.5 and $S \leq \alpha$, there exists a

finite ground instances set S_0 of S such that $S_0 \leq \alpha$. By Theorem 3.3, there exists a ground non-clausal multi-ary α-generalized resolution deduction from S_0 to α-$\square$. By Lifting Lemma (Theorem 4.6), there exists a non-clausal multi-ary α-generalized resolution deduction from S to α-$\square$.

The following two examples provide some illustration of completeness of non-clausal multi-ary α-generalized resolution deduction in LF(X).

Example 4.1 Let $\Lambda_9 = (L_9, \vee, \wedge, ', \rightarrow, a_1, a_9)$ be a Łukasiewicz implication algebra, b, c, d constants, x, y, w, r, s, t variables in $L_9F(X)$, $S = \{(P(s) \leftrightarrow Z_1(y)), (P(s) \leftrightarrow Z_1(y))' \vee (T(t) \rightarrow W(w)), ((T(t) \rightarrow W(w)) \rightarrow a_2) \vee (Z_2(s) \rightarrow R(r)), (Q(y) \rightarrow Z_1(b)) \vee (R(r) \rightarrow Z_1(c))', R(r) \rightarrow Z_1(d), (a_3 \rightarrow Q(y))'\}$. If we take the resolution level $\alpha = a_6$, then $S \leq \alpha$. By the completeness of non-clausal multi-ary a_6-generalized resolution (Theorem 4.9), there exists a non-clausal multi-ary a_6-generalized resolution refutation from S.

In fact, we take a ground substitution $\sigma = \{a / x, b / t, c / w, a / s, c / y, b / r\}$ of S, then $S^\sigma = \{(P(a) \leftrightarrow Z_1(c)), (P(a) \leftrightarrow Z_1(c))' \vee (T(b) \rightarrow W(c)), ((T(b) \rightarrow W(c)) \rightarrow a_2) \vee (Z_2(a) \rightarrow R(b)), (Q(c) \rightarrow Z_1(b)) \vee (R(b) \rightarrow Z_1(c))', R(b) \rightarrow Z_1(d), (a_3 \rightarrow Q(c))'\}$, and $S^\sigma \leq a_6$, then there exists a non-clausal multi-ary α-generalized resolution refutation ω^0 from S^σ in $L_9P(X)$ as follows:

(1) $P(a) \leftrightarrow Z_1(c)$
(2) $(P(a) \leftrightarrow Z_1(c))' \vee (T(b) \rightarrow W(c))$
(3) $((T(b) \rightarrow W(c)) \rightarrow a_2) \vee (Z_2(a) \rightarrow R(b))$
(4) $(Q(c) \rightarrow Z_1(b)) \vee (R(b) \rightarrow Z_1(c))'$
(5) $R(b) \rightarrow Z_1(d)$
(6) $(a_3 \rightarrow Q(c))'$
----
(7) $a_6 \vee (P(a) \leftrightarrow Z_1(c))' \vee (Z_2(a) \rightarrow R(b))$ by (2), (3)
(8) $a_6 \vee (R(b) \rightarrow Z_1(c))'$ by (4), (6)
----
(9) $a_6 \vee (Z_2(a) \rightarrow R(b))$ by (1), (7)
----
(10) a_6-$\square$ by (5), (8), (9)

By Lifting Lemma of non-clausal multi-ary α-generalized resolution, there exists a non-clausal multi-ary α-generalized resolution refutation ω from S in $L_9F(X)$, that is, we resume the variable symbols in S

which are substituted by σ in S^σ. Therefore, the non-clausal multi-ary α-generalized resolution refutation ω of S is:

$(P(s) \leftrightarrow Z_1(y))$,
$(P(s) \leftrightarrow Z_1(y))' \vee (T(b) \rightarrow W(c))$,
$((T(b) \rightarrow W(c)) \rightarrow a_2) \vee (Z_2(a) \rightarrow R(b))$,
$(Q(c) \rightarrow Z_1(b)) \vee (R(b) \rightarrow Z_1(c))'$,
$R(b) \rightarrow Z_1(d)$,
$(a_3 \rightarrow Q(y))'$,
$a_6 \vee (P(x) \rightarrow (P(x) \rightarrow Z_1(c))) \vee (Z_2(s) \rightarrow R(r))$,
$a_6 \vee (R(r) \rightarrow Z_1(d))'$,
$a_6 \vee (Z_2(s) \rightarrow R(r))$,
a_6-$\square$.

Furthermore, if we judge the α-unsatisfiability of S by α-resolution principle, we firstly transform S to its generalized conjunctive normal form $S_1 = \{P(s) \rightarrow Z_1(y), Z_1(y) \rightarrow P(s), (P(s) \rightarrow Z_1(y))' \vee (Z_1(y) \rightarrow P(s))' \vee (T(t) \rightarrow W(w)), ((T(t) \rightarrow W(w)) \rightarrow a_2) \vee (Z_2(s) \rightarrow R(r)), (Q(y) \rightarrow Z_1(b)) \vee (R(r) \rightarrow Z_1(c))', R(r) \rightarrow Z_1(d), (a_3 \rightarrow Q(y))'\}$.

Then we take a ground substitution $\sigma = \{a / x, b / t, c / w, a / s, c / y, b / r\}$ of S_1, then $S_1^\sigma = \{P(a) \rightarrow Z_1(c), Z_1(c) \rightarrow P(a), (P(a) \rightarrow Z_1(c))' \vee (Z_1(c) \rightarrow P(a))' \vee (T(b) \rightarrow W(c)), ((T(b) \rightarrow W(c)) \rightarrow a_2) \vee (Z_2(a) \rightarrow R(b)), (Q(c) \rightarrow Z_1(b)) \vee (R(b) \rightarrow Z_1(c))', R(b) \rightarrow Z_1(d), (a_3 \rightarrow Q(c))'\}$, and $S_1^\sigma \leq a_6$. We cannot get a binary α-resolution refutation of S_1^σ, but can get a relatively complex multi-ary α-resolution refutation [Xu *et al.* 2013] of S_1^σ, i.e., there exists a multi-ary α-resolution refutation ω^0 from S_1^σ in $L_9P(X)$ as follows:

(1) $P(a) \rightarrow Z_1(c)$
(2) $Z_1(c) \rightarrow P(a)$
(3) $(P(a) \rightarrow Z_1(c))' \vee (Z_1(c) \rightarrow P(a))' \vee (T(b) \rightarrow W(c))$
(4) $((T(b) \rightarrow W(c)) \rightarrow a_2) \vee (Z_2(a) \rightarrow R(b))$
(5) $(Q(c) \rightarrow Z_1(b)) \vee (R(b) \rightarrow Z_1(c))'$
(6) $R(b) \rightarrow Z_1(d)$
(7) $(a_3 \rightarrow Q(c))'$
----
(8) $a_6 \vee (Z_1(c) \rightarrow P(a))' \vee (T(b) \rightarrow W(c))$ by (1), (3)
----
(9) $a_6 \vee (T(b) \rightarrow W(c))$ by (2), (8)
----
(10) $a_6 \vee (Z_2(a) \rightarrow R(b))$ by (4), (9)
----

(11) $a_6 \vee (R(b) \to Z_1(c))'$ by (5), (10)

----

(12) a_6-□   by (6), (7), (11)

Similarly, by Lifting Lemma of multi-ary α-resolution, there exists a multi-ary α-resolution refutation ω from S_1 in $L_9F(X)$, that is, we resume the variable symbols in S_1 which are substituted by σ in S_1^σ. Therefore, the multi-ary α-resolution refutation ω of S_1 is:

$P(s) \to Z_1(y)$,
$Z_1(y) \to P(s)$,
$(P(s) \to Z_1(y))' \vee (Z_1(y) \to P(s))' \vee (T(t) \to W(w))$,
$((T(t) \to W(w)) \to a_2) \vee (Z_2(s) \to R(r))$,
$(Q(y) \to Z_1(b)) \vee (R(r) \to Z_1(c))'$,
$R(r) \to Z_1(d)$,
$(a_3 \to Q(y))'$,
$a_6 \vee (Z_1(y) \to P(s))' \vee (T(t) \to W(w))$,
$a_6 \vee (T(t) \to W(w))$,
$a_6 \vee (Z_2(s) \to R(r))$,
$a_6 \vee (R(r) \to Z_1(c))'$,
a_6-□.

Example 4.2 Let $(L_6, \vee, \wedge, ', \to, O, I)$ be an LIA defined as Example 2.4, a, c, $d \in L_6$, x, y, z, t, w, u variable symbols in $L_6F(X)$, $S = \{P(x), (P(x) \to d) \vee (Z_1(w) \leftrightarrow Q(y)), (Z_1(w) \leftrightarrow Q(y))' \vee (T(z) \to d), Z(t) \to T(z), (Z(t) \to Z_1(u))'\}$. If we take the resolution level $\alpha = d$, then $S \le d$. By the completeness of non-clausal multi-ary d-generalized resolution (Theorem 4.9), there exists a non-clausal multi-ary d-generalized resolution refutation from S.

In fact, we take a ground substitution $\sigma = \{a_1 / x, a_2 / y, a_3 / z, a_4 / t, a_5 / u, a_1 / w\}$ of S, where a_1, a_2, a_3, a_4, and a_5 are constant symbols in $L_6F(X)$, then $S^\sigma = \{P(a_1), (P(a_1) \to d) \vee (Z_1(a_1) \leftrightarrow Q(a_2)), (Z_1(a_1) \leftrightarrow Q(a_2))' \vee (T(a_3) \to d), Z(a_4) \to T(a_3), (Z(a_4) \to Z_1(a_5))'\}$, and $S^\sigma \le d$, then there exists a non-clausal multi-ary α-generalized resolution refutation ω^0 from S^σ in $L_6P(X)$ as follows:

(1) $P(a_1)$
(2) $(P(a_1) \to d) \vee (Z_1(a_1) \leftrightarrow Q(a_2))$
(3) $(Z_1(a_1) \leftrightarrow Q(a_2))' \vee (T(a_3) \to d)$
(4) $Z(a_4) \to T(a_3)$
(5) $(Z(a_4) \to Z_1(a_5))'$

----

(6) $d \vee (Z_1(a_1) \leftrightarrow Q(a_2))$ by (1), (2)

----

(7) $d \vee (T(a_3) \to d)$ by (3), (6)

----

(8) d-□   by (4), (5), (7)

By Lifting Lemma of non-clausal multi-ary α-generalized resolution, there exists a non-clausal multi-ary α-generalized resolution refutation ω from S in $L_6F(X)$ similar to Example 4.1. Therefore, the non-clausal multi-ary α-generalized resolution refutation ω of S is:

$P(a_1)$,
$(P(a_1) \to d) \vee (Z_1(w) \leftrightarrow Q(y))$,
$(Z_1(w) \leftrightarrow Q(y))' \vee (T(a_3) \to d)$,
$Z(a_4) \to T(a_3)$,
$(Z(a_4) \to Z_1(a_5))'$,
$d \vee Q(y)$,
$d \vee (T(z) \to d)$,
d-□.

Furthermore, if we judge the α-unsatisfiability of S by α-resolution principle, we firstly transform S to its generalized conjunctive normal form $S_1 = \{P(x), (P(x) \to d) \vee (Z_1(w) \to Q(y)), (P(x) \to d) \vee (Q(y) \to Z_1(w)), (Z_1(w) \to Q(y))' \vee (Q(y) \to Z_1(w))' \vee (T(z) \to d), Z(t) \to T(z), (Z(t) \to Z_1(u))'\}$.

Then we take a ground substitution $\sigma = \{a_1 / x, a_2 / y, a_3 / z, a_4 / t, a_5 / u, a_1 / w\}$ of S, then $S_1^\sigma = \{P(a_1), (P(a_1) \to d) \vee (Z_1(a_1) \to Q(a_2)), (P(a_1) \to d) \vee (Q(a_2) \to Z_1(a_1)), (Z_1(a_1) \to Q(a_2))' \vee (Q(a_2) \to Z_1(a_1))' \vee (T(a_3) \to d), Z(a_4) \to T(a_3), (Z(a_4) \to Z_1(a_5))'\}$, and $S_1^\sigma \le d$, then we cannot get a binary α-resolution refutation of S_1^σ, but can get a relatively complex multi-ary α-resolution refutation of S_1^σ [24], i.e., there exists a multi-ary α-resolution refutation ω^0 from S_1^σ in $L_6F(X)$ as follows:

(1) $P(a_1)$
(2) $(P(a_1) \to d) \vee (Z_1(a_1) \to Q(a_2))$
(3) $(P(a_1) \to d) \vee (Q(a_2) \to Z_1(a_1))$
(4) $(Z_1(a_1) \to Q(a_2))' \vee (Q(a_2) \to Z_1(a_1))' \vee (T(a_3) \to d)$
(5) $Z(a_4) \to T(a_3)$
(6) $(Z(a_4) \to Z_1(a_5))'$

14

(7) $d \vee (Z_1(a_1) \to Q(a_2))$ by (1), (2)

(8) $d \vee (Q(a_2) \to Z_1(a_1))$ by (1), (3)

----

(9) $d \vee (Q(a_2) \to Z_1(a_1))' \vee (T(a_3) \to d)$ by (4), (7)

----

(10) $d \vee (T(a_3) \to d)$ by (8), (9)

----

(11) $d\text{-}\square$ by (5), (6), (10)

Similarly, by Lifting Lemma of multi-ary α-resolution, there exists a multi-ary α-resolution refutation ω from S_1 in $L_6F(X)$, that is, we resume the variable symbols in S_1 which are substituted by σ in S_1^σ. Therefore, the multi-ary α-resolution refutation ω of S_1 is:

$P(x)$,
$(P(x) \to d) \vee (Z_1(w) \to Q(y))$,
$(P(x) \to d) \vee (Q(y) \to Z_1(w))$,
$(Z_1(w) \to Q(y))' \vee (Q(y) \to Z_1(w))' \vee (T(z) \to d)$,
$Z(t) \to T(z)$,
$(Z(t) \to Z_1(u))'$
$d \vee (Z_1(w) \to Q(y))$,
$d \vee (Q(y) \to Z_1(w))$,
$d \vee (Q(y) \to Z_1(w))' \vee (T(z) \to d)$,
$d \vee (T(z) \to d)$,
$d\text{-}\square$.

Theorem 4.8 and Theorem 4.9 show that the non-clausal multi-ary α-generalized resolution deduction in LF(X) is sound and complete, along with Examples 4.1 and 4.2 to illustrate its advantages in terms of reasoning capability and efficiency. Similarly in LP(X), Examples 4.1 and 4.2 show that for an α-unsatisfiable set of general g-clause in LF(X), if we convert it into respectively generalized conjunctive normal form, and judge it by α-resolution principle, then it may not lead to a binary α-resolution refutation partially because its restriction of the number of α-resloved literals is 2. Although sometimes it may lead to a multi-ary α-resolution refutation, but it is more complex than the non-clausal multi-ary α-resolution refutation.

Consequently, the proposed work is a great extension of the results in [24-26] in terms of soundness and completeness, applicability, reasoning capability and reasoning efficiency.

Remark 4.2: In fact, it follows from Theorem 4.5 that the determination of α-generalized resolution in LF(X) can be equivalently transformed into that of α-generalized resolution in LP(X) to some extents, which reduces the difficulty of α-generalized resolution in LF(X) to some extents. Hence, the determination of α-generalized resolution in LP(X) would be the next key step for developing efficient α-resolution reasoning algorithm for LP(X) as well as LF(X). However, similar to the one indicated in [24], the practical implementations of a resolution deduction algorithm are much more complex, especially in the case of first-order logic. It can be tracked back to 1931 when Godel proposed the famous undecidability theory, that is, it is impossible to construct a single algorithm that can always lead to a correct true-or-false answer for all logical formulae in a specified deductive system.

The focus of this paper is on the resolution principle and the theoretical soundness and completeness of this resolution-based automated deduction, not on the concrete algorithms or search strategies for implementation. For resolution-based automated reasoning in lattice-valued logic based on LIA, 1) it is more complex than that in classical logic from the logical point of view; 2) it will be not that straightforward either in determining or search which group of generalized literals could be α-resoluble, that is, resoluble at a truth-value level α, or determining at least how many generalized literals can be chosen in the α-resolution group once given a truth-value level α; 3) although the resolution process can borrow the similar ideas from classical logic, it becomes more complex due to the more complex generalized literals involved in the resolution and also the fact that it allows the choice of various truth-value level resolution (different from the only case of α=O in classical logic).

Consequently, it is much harder or more computationally complex to achieve the α-resolution-based automated reasoning algorithm in lattice-valued logic based on LIA than to achieve it in the classical logic. This kind of concrete algorithms or search strategies for implementation will be still one of challenge problems in α-resolution-based automated reasoning in lattice-valued logic based on LIA, will need more efforts to investigate in the future, this topic,

however, is beyond the scope of the present work.

5. Related works

Lattice-valued logics as ones of the most important many-valued logics, extend the chain-type truth-valued field to a general lattice structure in which the truth-values are incompletely comparable with each other [23, 34-43]. Lattice-valued logics are thus an important and promising research direction that provides an alternative logical approach to dealing with imprecision and incomparability as well [23].

Up to now, many researchers have made investigation on resolution-based automated reasoning in the framework of fuzzy logic and many-valued logics, and obtained some important results [38, 42, 44-80].

Aiming at establishing automated deduction for many-valued logic, Xu *et al.* introduced a binary resolution at a certain truth-value level α (called *α-resolution principle*) and developed the α-resolution deduction in a lattice-valued logic L(X) based on a lattice-valued logic algebra – lattice implication algebra (LIA) and proved its soundness and completeness [25, 26]. Compared with the resolution principle in Boolean logic, the α-resolution principle in lattice-valued logic L(X) has new features such as: (a) α-resolution is proceeded at different truth-value level α (with the possible incomparability) chosen from the truth-valued field — LIA; (b) α-resolution is based on generalized literals, which contain constant formula and more general implication connective than the one in the classical logic. Hence the expressive power is enhanced. Actually, implication connectives in L(X) are not reducible to other classical logical connectives, which is different from the Kleene implication (i.e., $p \rightarrow q = p' \vee q$). This irreducibility is semantically meaningful, complicates the calculus; (c) judging whether two generalized literals are α-resolvable should consider both semantic and syntax consistently [25, 26].

Although LIAs have been investigated independently, it has been proved [23] that LIAs are categorically equivalent to (i.e., coincide with) the class of MV algebra, which are the algebraic semantics of Łukasiewicz logic. LIAs form a variety of algebras and the variety of LIA-algebras contains all Boolean algebras and Łukasiewicz algebra (i.e., the variety of algebras of Lukasiewicz logic), two of the most commonly investigated logic algebra in classical logic and many-valued logic. The focus of the present paper is establishing a sound and complete resolution-based reasoning system based on LIAs, which means establishing a sound and complete resolution-based reasoning system based on Boolean algebra and also Łukasiewicz algebra at least. The results obtained in lattice-valued logics L(X) based on LIA in several ways have extended and expanded Pavelka fuzzy logic [23, 36]. This shows that the investigation of resolution-based automated reasoning for L(X) is worthwhile and is an important extension of classical logic and also some many-valued logics, and is of importance to the research and practitioner community in automated reasoning (where these ideas can be applied in some other relevant logic systems based on different logic semantic algebras). This reflects the key motivation for the proposed work.

Although there has been some research work on resolution-based automated reasoning methods based on non-classical logic (e.g., for fuzzy logic and many-valued logic) as cited earlier, the essential idea in many of those methods is to transform the resolution algorithm in fuzzy logic and many-valued logic to that of classical logic, because there is no constant formula involved in the syntax of the logic system so they have the syntactical equivalence, this is one of the key differences form the automated deduction in signed logic or annotated logic [20, 44, 56, 57, 81, 82], Bilattice-based logics [83], probabilistic logic [84], similarity-based logic [85]. The works related to Lukasiewicz logic have been mainly focused on generalized CNF based on bold product and bold sum operators or logic programming [59, 76, 77]. As far as we know, proof theory for lattice-valued logic has so far not been extensively developed.

6. Conclusions

In this paper, a *non-clausal multi-ary α-generalized resolution principle* and *its resolution deduction* for lattice-valued logic based on a lattice-valued logical algebra - LIA were proposed. The definitions of the

general form of non-clausal multi-ary α-generalized resolution and non-clausal multi-ary α-generalized resolution deduction in LP(X) were given, along with its soundness and completeness. In order to obtain the completeness of this general form of non-clausal multi-ary α-generalized resolution principle in LF(X), the α-unsatisfiability of the general form of logical formula was discussed. Finally, by Lifting Lemma and the completeness of the ground case, the completeness of the general case was obtained. This contribution is expected to provide a theoretical foundation for more efficient and effective resolution based automated reasoning algorithms and tools in lattice-valued logic with the goal of applying them to some practical fields such as expert system design, intelligent robot design, and machine learning system design under uncertain environment. The further research will be concentrated on contriving an algorithm to achieve the efficiency of the non-clausal multi-ary α-generalized resolution, and investigating the non-clausal multi-ary α-generalized resolution in linguistic truth-valued lattice-valued logic[86] for some applications.

Acknowledgments

References

1. J.P. Robinson, A machine-oriented logic based on the resolution principle, J. ACM 12 (1965), 23-41.
2. Johan van Benthem and Alice ter Meulen, Handbook of Logic and Language, 2rd edition, Elsevier, 2010.
3. W. Bibel, Early history and perspectives of automated deduction. Proc. of the 30th Annual German Conference on AI, Osnabruck, Germany, September 10-13, 2007, pages 2–18.
4. [Davis 2001] M. Davis, The early history of automated deduction, in: A. Robinson and A. Voronkov eds.: "*Handbook of Automated Reasoning*", pages: 5-15.
5. D. Plaisted, History and prospects for first-order automated deduction, *Proceedings of the 25th International Conference on Automated Deduction*, Berlin, Germany, August 1-7 2015, pp. 3-28.
6. S.H. Muggleton, Alan Turing and the development of Artificial Intelligence, AI communications, forthcoming, 2013.
7. S.H. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. Journal of Logic Programming, 19, 20 (1994), 629–679.
8. S.H. Muggleton, L. De Raedt, D. Poole, I. Bratko, P. Flach, and K. Inoue. ILP turns 20: biography and future challenges. Machine Learning, 86(1) (2011), 3–23.
9. S. H. Lee, N. Van, Danny, Yang, Kyung-Ae Lee, In-Hee Zhang, B. T. Park, H. Tai. Biomolecular theorem proving on a chip: a novel microfluidic solution to a classical logic problem, Lab Chip, 12 (2012), 1841-1848.
10. S. G. Naveen, L. John, B. Selmer, Small Steps toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation, Unconventional Computation and Natural Computation, 7956 (2013), 102-112.
11. C. Bouillaguet, V. Kuncak, T. Wies, K. Zee, and M. Rinard. Using first-order theorem provers in the Jahob data structure verification system. In: Verification, Model Checking, and Abstract Interpretation (pp. 74-88). Springer Berlin Heidelberg.
12. B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. In: Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on (pp. 331-340). IEEE.
13. C. Green, The application of theorem proving to question-answering systems Ph.D. thesis, Stanford University, Stanford, 1969.
14. U. Furbach, I. Glöckner, and B. Pelzer, An application of automated reasoning in natural language question answering, AI Communications, 23 (2-3)(2010), 241-265.
15. Y. Xu, J. Liu, D. Ruan, and T.T. Lee, On the consistency of rule-bases based on lattice–valued first-order logic LF(X), International Journal of Intelligent System, 21 (2006), 399-424.
16. X.H. Wang, X.H. Liu, Generalized resolution, Chinese journal of computers, 2 (1982), 81-92 (in Chinese).
17. Z. Stachniak, Resolution Proof Systems: An Algebraic Theory, Kluwer Academic Publisher, Netherlands, 1996.
18. Z. Stachniak, Non-Clausal Reasoning with Definite Theories, Fundamenta Informaticae, 48 (2001), 1–26.
19. N. Murray, Completely non-clausal theorem proving. Artificial Intelligence, 18 (1982), 67-85.
20. R. Hahnle, Automated deduction in multiple-valued logics, 1993, Oxford University Press, Inc.
21. H. Habiballa, Non-clausal resolution - theory and

practice. Research report: University of Ostrava, 2000.

22. Y. Xu, Lattice implication algebras, J. Southwest Jiaotong University, 89(1) (1993), 20-27 (in Chinese).

23. Y. Xu, D. Ruan, K.Y. Qin, and J. Liu, Lattice-Valued Logic: An Alternative Approach to Treat Fuzziness and Incomparability, Springer-Verlag, Berlin, 2003.

24. Y. Xu, J. Liu, X.M. Zhong, and S.W. Chen, Multi-ary α-resolution principle for a lattice-valued logic, IEEE Transactions on Fuzzy Systems, 21 (5) (2013), 898–912.

25. Y. Xu, D. Ruan, E.E. Kerre, and J. Liu, α-Resolution principle based on lattice-valued propositional logic LP(X). Information Sciences, 130(2000), 195-223.

26. Y. Xu, D. Ruan, E.E. Kerre, and J. Liu, α-Resolution principle based on lattice-valued first-order lattice-valued logic LF(X), Information Science, 132 (2001), 221-239.

27. Y. Xu, W. T. Xu, X. M. Zhong and X. X. He, α-Generalized Resolution Principle Based on Lattice-Valued Propositional Logic System LP(X), in Proc. of the 9th International FLINS Conference on Foundations and Applications of Computational Intelligence (FLINS2010), 66–71.

28. Y. Xu, J. Liu, D. Ruan, and X.B. Li, Determination of α-resolution in lattice-valued first-order logic LF(X), Information Sciences, 181 (2011), 1836–1862.

29. Y Xu, J. Liu, X.X. He, X.M. Zhong, and S.W. Chen, Non-clausal multi-ary α-generalized resolution principle for a lattice-valued propositional logic, Proceeding of the 11th International FLINS Conference on Decision Making and Soft Computing (FLINS2014)/9th International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2014), Joao Pessoa, Brazil, August 17-20, 2014, pp.

30. Y Xu, J. Liu, X.X. He, X.M. Zhong, and S.W. Chen, Non-clausal multi-ary α-generalized resolution principle for a lattice-valued first-order logic, IEEE Proceeding of the 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE2015), 24-27 Nov. 2015, Taipei, Taiwan, pp. 1-7.

31. W.T. Xu, W.Q. Zhang, D.X. Zhang, Y. Yang, X.D. Pan, α -Resolution method for lattice-valued Horn generalized clauses in lattice-valued propositional logic systems, International Journal of Computational Intelligence Systems, 2015(8): pp 75-84.

32. X.M. Zhong, Y. Xu, J. Liu, and S.W. Chen, General form of alpha-resolution based on linguistic truth-valued lattice-valued logic, Soft Computing Journal, Vol. 16, 2012, pp. 1767-1781.

33. Y. Liu, Y. Xu, 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, 29(4)(2015): pp 1581-1593.

34. J. A. Goguen, L-type Fuzzy Sets, J. Math. An. Appl., 18 (1967), 145-174.

35. J. A. Goguen, The logic of inexact concepts, Synthese, 19(1968/69), 325-373.

36. J. Pavelka, On fuzzy logic I: Many-valued rules of inference, II: Enriched residuated lattices and semantics of propositional calculi, III: Semantical completeness of some many-valued propositional calculi, in: Zeitschr. F. Math. Logik und Grundlagend. Math, 25(1979), 45-52, 119-134, 447-464.

37. S. Gottwald, A Treatise on Many-Valued Logics. Studies in Logic and Computation, Vol. 9, Research Studies Press Ltd., Baldock, 2001.

38. L. Bolc, P. Borowik, Many-Valued Logics, Springer, Berlin, 1992.

39. V. Novak. Fuzzy Sets and Their Applications. Bristol: Adam Hilger, 1989.

40. V. Novak, I. Perfilieva, and J. Mockor. Mathematical Principles of Fuzzy Logic, Dordrecht: Kluwer, 2000.

41. P. Hajek, Metamathematics of fuzzy logic. Kluwer Academic Publishers - Dordrecht, 2000.

42. R. Cignoli, I. D'Ottaviano, and D. Mundici. Algebraic Foundations of Many-valued Reasoning, Dordrecht: Kluwer, 2000.

43. G. Birkhoff, Lattice Theory, 3rd edition, American Mathematical Society, Providence, R.L., 1967.

44. C.C. Chang, Algebraic analysis of many valued logic. Trans. Amer. Math. Soc., 88 (1958), 467–490.

45. R. Hahnle, Many-valued logic and mixed integer programming, Annals of Mathematics and Artificial Intelligence 12 (3-4)(1994), 231-263

46. P. Kullmann, S. Sandri, An annotated logic theorem prover for an extended possibilistic logic, Fuzzy Sets and Systems 144 (1) (2004) 67–91.

47. D. Guller, Binary resolution over Boolean lattices, Fuzzy Sets and Systems 157(2006), 2100-2127.

48. D. Guller, Binary resolution over complete residuated stone lattices, Fuzzy Sets and Systems, 159(2008), 1031-1041.

49. J. Ignjatović, M. Ćirić, and S. Bogdanović, Determinization of fuzzy automata with membership values in complete residuated lattices, Information Sciences, 178 (1)(2008), 164-180.

50. D. Guller, On the refutational completeness of signed binary resolution and hyperresolution, Fuzzy Sets and Systems, 160 (8)(2009), 1162-1176.

51. M. Banibrata and S. Raha. Approximate reasoning in fuzzy resolution, Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American. IEEE.

52. M. Banibrata and S. Raha, Fuzzy resolution with

similarity-based reasoning. Recent Developments and New Directions in Soft Computing. Springer International Publishing, 2014, pp. 361-378.

53. C. Ansótegui, M. Bofill, F. Manyà, & M. Villaret, SAT and SMT technology for many-valued logics. Journal of Multiple-Valued Logic & Soft Computing, 24(1-4)(2015), pp. 151-172.

54. C. Ansótegui, M. Bofill, F. Manyà, & M. Villaret, Automated theorem provers for multiple-valued logics with satisfiability modulo theory solvers, Fuzzy Sets and Systems, 292(1)(2016), pp. 32–48.

55. X.H. Liu, Resolution-Based Automated Reasoning, Academic Press, Beijing, China, 1994 (in Chinese).

56. V. Loia, S. Senatore and M.I. Sessa, Similarity-based SLD resolution and its role for web knowledge discovery, Fuzzy Sets and Systems, 144 (1)(2004), 151-171.

57. J.J. Lu, N.V. Murray, E. Rosenthal, A framework for automated reasoning in multiple-valued logics, Journal of Automated Reasoning 21 (1998) 39–67.

58. J. Medina and M. Ojeda-Aciego, Multi-adjoint t-concept lattices, Information Sciences, 180 (5) (2010), 712-725.

59. S. Lehmke. A resolution-based axiomatisation of `bold' propositional fuzzy logic. In: Fuzzy Sets, Logics, and Reasoning about Knowledge (Eds. by D. Dubois, E. P. Klement, and H. Prade), Kluwer Academic Publishers, Applied Logic, 1999.

60. C.S. Kim, D.S. Kim, J.S. Park, A new fuzzy resolution principle based on the antonym, Fuzzy Sets and Systems 113 (2) (2000) 299–307.

61. J. Liu, D. Ruan, Y. Xu, Z.M. Song, A resolution-like strategy based on a lattice-valued logic, IEEE Transactions on Fuzzy Systems, 11 (4) (2003), 560-567.

62. J. Hoffmann, Finding a tree structure in a resolution proof is NP-complete, Theoretical Computer Science, 410 (21-23)(2009), 2295-2300.

63. L. Di Lascio and A. Gisolfi, Graded tableaux for Rational Pavelka Logic, International Journal of Intelligent Systems, 20(12) (2005), 1273–1285.

64. P. Hofner and G. Struth, Automated reasoning in kleene algebra, In CADE 2007, F. Pfennig (ed.), LNCS 4603, 2007, pp. 279-294.

65. X.X. He, Y. Xu, J. Liu, D. Ruan, α-Lock resolution method for lattice-valued logic based on lattice implication algebra, Engineering Applications of Artificial Intelligence, Vol. 24, 2011, 1274-1280.

66. X.X. He, J. Liu, Y. Xu, L. Martinez, and D. Ruan, On α-satisfiability and its α-Lock resolution in a finite lattice-valued logic, Logic Journal of the IGPL, Logic Journal of IGPL, 20(3)(2011), 579–588.

67. X.X. He, Y. Xu, J. Liu, S.W. Chen, On compatibilities of α-lock resolution method in linguistic truth-valued lattice-valued logic, Soft Computing Journal, 16(4)(2012), pp. 699-709.

68. X.X. He, J. Liu, Y. Xu, L. Martinez, and D. Ruan, On α-satisfiability and its α-Lock resolution in a finite lattice-valued logic, Logic Journal of the IGPL, 20(3)(2012), pp. 579-588.

69. X.X. He, Y. Xu, J. Liu, S.W. Chen. A unified algorithm for finding k-IESFs in linguistic truth-valued lattice-valued propositional logic, Soft Computing, 18(11)(2014), pp. 2135-2147.

70. X.X. He, Y. Xu, J. Liu, α-Lock paramodulation for lattice-valued propositional logic, International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2015), 24 Nov – 27 Nov, 2015, Taibei, pp 18-20

71. X.M. Zhong, Y. Xu, J. Liu, S.W. Chen, α-Quasi-lock semantic resolution method based on lattice-valued logic. International Journal of Computational Intelligence Systems, 7(3)(2014): 418-431.

72. X.M. Zhong and Y. Xu, alpha- group quasi-lock semantic resolution method based on lattice-valued propositional logic LP(X). Multiple-Valued Logic and Soft Computing 22(4-6)(2014): 581-598.

73. H.R. Jia, Y. Xu, Y. Liu, and J. Liu, Alpha-minimal resolution principle for a lattice-valued logic, International Journal of Computational Intelligence Systems, Vol. 8. No. 1, 2015, pp. 34-43.

74. S. Schockaert, J. Janssen, and D. Vermeir, Satisfiability checking in Łukasiewicz logic as finite constraint satisfaction, Journal of Automated Reasoning, 49(2012), pp. 493–550.

75. D. Smutná-Hliněná and P. Vojtáš, Graded many-valued resolution with aggregation, Fuzzy Sets and Systems, 143 (2004), 157-168.

76. V. Sofronie-Stokkermans, Automated theorem proving by resolution for finitely-valued logics based on distributive lattices with operators, Multiple-Valued Logic—An International Journal, 6(3-4) (2001), 289-344.

77. V. Sofronie-Stokkermans and C. Ihlemann, Automated reasoning in some local extensions of ordered structures, Journal of Multiple-Valued Logic and Soft Computing, 13(4-6) (2007), 397-414.

78. F. Liu, Tabulation proof procedures for fuzzy linguistic logic programming, International Journal of Approximate Reasoning, 63(2015), pp. 62-88.

79. J. Ma, W.J. Li, D. Ruan, Y. Xu, Filter-based resolution principle for lattice-valued propositional logic LP(X), Information Sciences, 177 (2007), 1046-1062.

80. T.D. Khang, A resolution method for linguistic many-valued logic. Applied Mathematics &

Information Sciences, 7(3)(2013), pp. 1193.

81. M. Kifer and V.S. Subrahmanian, Theory of generalized annotated logic programming and its applications, Journal of Logic Programming, 12(4)(1992): 335-367.

82. E. Komendantskaya, A.K. Seda, Sound and complete SLD-resolution for bilattice-based annotated logic programs, Electronic Notes in Theoretical Computer Science 2009, 225, 141-159.

83. M. Fitting, Bilattices and the semantics of logic programming. Journal of Logic Programming, 11(1991): 91-116.

84. F. Bacchus, Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities, 1991, MIT Press Cambridge, MA, USA.

85. P. Julián-Iranzo and C. Rubio-Manzano, A sound and complete semantics for a similarity-based logic programming language, Fuzzy Sets and Systems 317(15)(2017): 1–26.

86. Z. Pei, D. Ruan, J. Liu, Y. Xu, Linguistic Values-based Intelligent Information Processing: Theory, Methods, and Applications, Atlantis Press, 2009.