International Journal of Computational Intelligence Systems, Vol. 5, No. 6 (November, 2012), 1120-1134

α-Generalized lock resolution method in linguistic truth-valued lattice-valued logic

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

[1] *Intelligent Control Development Center, Southwest Jiaotong University, Chengdu 610031, Sichuan, PR China*

E-mail: xinghe010@gmail.com, xuyang@home.swjtu.edu.cn

[2] *School of Computing and Mathematics, University of Ulster, Northern Ireland, UK*

E-mail: j.liu@ulster.ac.uk, Chen-S1@email.ulster.ac.uk

Received 17 February 2012

Accepted 31 July 2012

Abstract

This paper focuses on efficient non-clausal resolution-based automated reasoning methods and algorithms for a lattice-ordered linguistic truth-valued logic, which corresponds to extensions of α-lock resolution. Firstly, α-generalized lock resolution is proposed for lattice-valued propositional logic and first order logic, respectively, along with their concepts, soundness and completeness. Then, α-generalized lock resolution for first order linguistic truth-valued lattice-valued logic $\mathscr{L}_{V(n\times2)}$F(X) is equivalently transformed into that for propositional logic L_nP(X), which can greatly reduce the complexity of the resolution procedure. Finally, α-generalized linear semi-lock resolution is discussed, and its general algorithm is also contrived. This work provides more efficient and natural resolution automated reasoning scheme in linguistic truth-valued logic based on lattice implication algebra with the aim at establishing formal tools for symbolic natural language processing.

Keywords: Non-clausal resolution; α-Generalized lock resolution; α-Generalized linear semi-lock resolution; Linguistic truth-valued lattice-valued logic; Lattice implication algebras

1. Introduction

In mathematical logic and automatic theorem proving, resolution principle, proposed by Robinson [19], is a rule of inference leading to a refutation theorem-proving technique for sentences in first order logic. Most conventional resolution methods [3,4] convert a theorem into its clausal form before attempting to find a proof, such a translation often obscures the structure of original formula, and may even increase the length of the formula by an exponential amount in the worst case [17,24]. Generalized resolution [10,11] is one of non-clausal resolution methods to describe and deal with the complex problems more naturally, which attempts to reason by using formulae directly without translating them to clausal forms. Similar ideas for non-clausal resolution have also been proposed by Murray [13], and further developed in [1,2,8,14,17,20,21,22], as well as some refined resolution methods for improving the efficiency of non-clausal resolution investigated and reported in [23,25]. Meanwhile, the completeness of these methods has been

*Corresponding author.

purposed in classical first order logic. Therefore, it increases the attractiveness of resolution based automated reasoning for practical applications such as program verification, query-answering system, problem of solving, programming, etc.

Uncertainty is often associated with human's intelligent activities, it is rather difficult to represent and reason it only by numbers or symbols in classical logic. Inspired with the idea of generalized resolution and by combining it with α-resolution [28,29], Xu et al. [30,32,34] proposed α-generalized resolution in lattice-valued logic [18,27] based on lattice implication algebra (LIA) [26], which extends the chain type truth-valued field to general lattice in which the truth-values are incompletely comparable with each other. Hence, it provides a naturally formal framework [9] to represent and reason uncertain knowledge especially for incomparability. However, the process of α-generalized resolution is level saturated if no refined strategy is restricted, and many redundant clauses inevitably generate, thus it prevents providing a universal procedure for finding the refutation faster.

α-Lock resolution [5,6,7] is a simple, but effective refinement on α-resolution, it is an α-resolution of locked generalized clauses in which the generalized literals resolved upon have the minimal indices in their respective generalized clauses, and the generalized literals of resolvent inherit the indices they had in their parents. It can significantly reduce the production of redundant generalized clauses. In this paper, we intend to restrict lock resolution strategy on α-generalized resolution, and propose α-generalized lock resolution method. Compared with α-lock resolution [7], α-generalized lock resolution can validate α-unsatisfiability of logical formulae without converting them to according generalized clausal forms, this simplification avoids the clausal transforming process. More remarkably, it is a dynamic resolution, i.e., the number of resolved literals is not limited to 2, but allowed to resolve in batch. In fact, in most cases, many generalized literals can be α-resolved simultaneously, but not α-resolved if only two generalized literals are taken from its subsets. The resolution of the generalized form is much easier to preserve its completeness and therefore makes its procedures be applicable to more logical formulae. Also, bathing resolution makes many generalized clauses be involved and more generalized literals be deleted in one resolution step, hence it enhances the efficiency of α-generalized resolution to some extent. Furthermore, for improving the efficiency of α-generalized lock resolution, we propose α-generalized linear semi-lock resolution by combining it with α-generalized linear resolution. However, such a combination does not preserve its completeness, hence we discuss it in a practical logic system and under some conditions for lock index assignments.

Linguistic truth-valued lattice-valued first order logic ($\mathscr{L}_{V(n\times 2)}$F(X)) [16,30] is an appropriate logic system for qualitatively representing and reasoning linguistic-values based information in natural language [35,36]. Its valuation field, linguistic truth-valued LIA [30], is a logical algebraic structure partially ordered with linguistic truth values. It has many unique characters such as linguistic truth values adopted having apparent distinguish ability, in accordance with the meaning of commonly used natural language and covering commonly used natural linguistic expressions in real life. Hence, it provides a formalism for the development of logic system based on linguistic truth values and resolution based automated reasoning in linguistic truth-valued logic system as well. Properties of α-resolution fields, weak completeness and equivalent transformations have been highlighted in [30,33], as well as applications investigated in [12,31].

For studying resolution methods in $\mathscr{L}_{V(n\times 2)}$F(X), two potential ways need to be considered. One is performing reasoning directly in this logic system. Of course, it is a natural way to process the linguistic-values based information without losing any information, but the operations in its valuation field $\mathscr{L}_{V(n\times 2)}$ are defined by means of isomorphic mappings and different results for different meta truth values, and its elements are binary arrays with linguistic truth values, hence it is relatively complex to get the truth-values of formulae in $\mathscr{L}_{V(n\times 2)}$F(X). The other is equivalently or conditionally transforming the resolution methods from $\mathscr{L}_{V(n\times 2)}$F(X) into those in some simpler logic systems, and therefore the operations in them are easier to performed, then the computational complexity can be simplified accordingly. Meanwhile, to preserve their completeness, the structures of logical formulae should not be changed in the transformations. Hence, it should be a good alternative for researching the resolution methods in $\mathscr{L}_{V(n\times 2)}$F(X). With this in mind, this paper transforms α-generalized lock resolution in $\mathscr{L}_{V(n\times 2)}$F(X) into that in L_nP(X) whose truth-valued domain is a Łukasiewicz implication algebra on a finite chain L_n, and discusses α-generalized linear semi-lock resolution in L_nP(X), which can further improve the efficiency of α-generalized resolution.

The paper is organized as follows. Section 2 gives some preliminary relevant concepts about α-resolution and α-generalized resolution principle in lattice-valued logic based on LIA. In Section 3, α-generalized lock resolution method is introduced, and its soundness and completeness are obtained. In Section 4, α-generalized lock resolution is transformed between $\mathscr{L}_{V(n\times 2)}$F(X) and L_nP(X). Section 5 discusses the α-generalized linear semi-lock resolution method which can further improve the efficiency of α-generalized lock resolution. The paper concludes in Section 6.

2. Preliminaries

2.1. *α-Resolution principle in lattice-valued logic based on LIA*

Definition 1. [26,30] 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 $\rightarrow: L\times L \longrightarrow L$ a mapping. $\mathscr{L} = (L,\vee,\wedge,',\rightarrow,O,I)$ 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) $(x \wedge y) \rightarrow z = (x \rightarrow z) \vee (y \rightarrow z)$.

In order to deal with quantifiers, in what follows, we suppose that L is a complete lattice.

Proposition 1. [26,30] *(Łukasiewicz implication algebra on a finite chain L_n) Let L_n be a finite chain, $L_n = \{a_i | 1 \leqslant i \leqslant n\}$ and $a_1 < a_2 < \ldots < a_n$, define for any $a_i, a_j \in L_n$, $a_i \vee a_j = a_{max(i,j)}$, $a_i \wedge a_j = a_{min(i,j)}$, $(a_i)' = a_{n-i+1}$, $a_i \rightarrow a_j = a_{min(n-i+j,n)}$, then $\mathscr{L}_n = (L_n,\vee,\wedge,',\rightarrow,a_1,a_n)$ is an LIA.*

All elements in Łukasiewicz implication algebra are completely comparable.

Proposition 2. [16,31] *Let $L_n = \{a_1,\ldots,a_n\}$, $a_1 < a_2 < \ldots < a_n$, $L_2 = \{b_1,b_2\}$, $b_1 < b_2$, $(L_n,\vee,\wedge,',\rightarrow,a_1,a_n)$ and $(L_2,\vee,\wedge,',\rightarrow,b_1,b_2)$ be two Łukasiewicz implication algebras. The Hasse diagram of $L_n \times L_2$ is depicted in Figure 1, and for any (a_i,b_j), $(a_k,b_l) \in L_n \times L_2$, define $(a_i,b_j) \vee (a_k,b_l) = (a_i \vee a_k, b_j \vee b_l), (a_i,b_j) \wedge (a_k,b_l) = (a_i \wedge a_k, b_j \wedge b_l)$, $(a_i,b_j)' = (a_i', b_j')$, $(a_i,b_j) \rightarrow (a_k,b_l) = (a_i \rightarrow a_k, b_j \rightarrow b_l)$, then $\mathscr{L}_{n\times 2} = (L_n \times L_2,\vee,\wedge,',\rightarrow,(a_1,b_1),(a_n,b_2))$ is an LIA.*

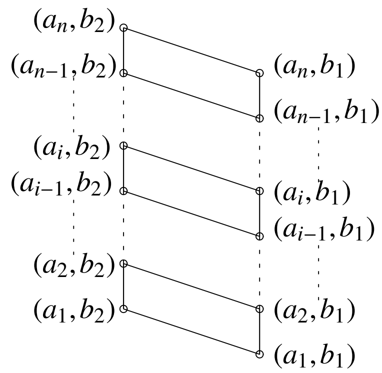

Figure 1: Hasse Diagram of $L_n \times L_2$

Definition 2. [16,31] Let $AD_n = \{h_1,h_2,\ldots,h_n\}$ be a set with n linguistic modifiers and $h_1 < h_2 < \ldots < h_n$, $MT = \{f,t\}$ be a set of meta truth values, and $f < t$. Denote $L_{V(n\times 2)} = AD_n \times MT$. Define a mapping g as $g : L_{V(n\times 2)} \longrightarrow L_n \times L_2$,

$$g((h_i,mt)) = \begin{cases} (a_i', b_1), & mt = f, \\ (a_i, b_2), & mt = t. \end{cases}$$

Then g is bijection, denote its inverse mapping as g^{-1}. For any $x, y \in L_{V(n\times 2)}$, define $x \vee y = g^{-1}(g(x) \vee g(y))$, $x \wedge y = g^{-1}(g(x) \wedge g(y))$, $x' = g^{-1}(g(x)')$, $x \to y = g^{-1}(g(x) \to g(y))$, then $\mathscr{L}_{V(n\times 2)} = (L_{V(n\times 2)}, \vee, \wedge, ', \to, (h_n, f), (h_n, t))$ is called a linguistic truth-valued LIA generated by AD_n and MT, its elements are called linguistic truth values, and g is an isomorphic mapping from $\mathscr{L}_{V(n\times 2)}$ to $\mathscr{L}_{n\times 2}$.

Definition 3. [28] Let X be a set of propositional variables, $T = L \cup \{', \to\}$ be a type with ar($'$) = 1, ar($\to$) = 2 and ar(a) = 0 for every $a \in L$. The propositional algebra of the lattice-valued propositional calculus on the set X of propositional variables is the free T algebra on X and is denoted by LP(X).

Remark 1. Specially, when the field with valuation of LP(X) is an $\mathscr{L}_{V(n\times 2)}$, this specific LP(X), i.e., $\mathscr{L}_{V(n\times 2)}P(X)$, is a linguistic truth-valued lattice-valued propositional logic system. Similarly, the truth-valued domain of $L_nP(X)$ is a Łukasiewicz implication algebra $\mathscr{L}_n$.

Definition 4. [28] A valuation of LP(X) is a propositional algebra homomorphism γ: LP(X) $\longrightarrow L$.

Definition 5. [28] Let F be a logical formula in LP(X), $\alpha \in L$. If there exists a valuation γ_0 of LP(X) such that $\gamma_0(F) \geqslant \alpha$, F is satisfiable by a truth-value level α, in short, α-satisfiable. If $\gamma(F) \geqslant \alpha$ for every valuation γ of LP(X), F is valid by the truth-value level α, in short, α-valid. If $\gamma(F) \leqslant \alpha$ for every valuation γ of LP(X), F is always false by the truth-value level α, in short, α-false.

Definition 6. [28] A logical formula F in LP(X) is called an extremely simple form, in short ESF, if a logical formula F^* obtained by deleting any constant or literal or implication term appearing in F is not equivalent to F.

Definition 7. [28] A logical formula F in LP(X) (i.e., $F \in$ LP(X)) is called an indecomposable extremely simple form, in short IESF, if

(1) F is an ESF containing connectives $\to$ and $'$ at most.

(2) For any $G \in$ LP(X), if $G \in \overline{F}$ in $\overline{LP(X)}$, then G is an ESF containing connectives $\to$ and $'$ at most.

Definition 8. [28] All the constants, literals and IESFs in LP(X) are called generalized literals. A disjunction of finite generalized literals is called a generalized clause.

The truth-value domain of lattice-valued first-order logic LF(X) is an LIA. This logic system can be used to deal with propositions with quantifiers [29,30].

Remark 2. Similar to the notation of $\mathscr{L}_{V(n\times 2)}P(X)$, the truth-valued domain of first order logic systems $\mathscr{L}_{V(n\times 2)}F(X)$ and $L_nF(X)$ are $\mathscr{L}_{V(n\times 2)}$ and $\mathscr{L}_n$, respectively.

Definition 9. [29] A logical formula G in LF(X) is a g-literal, if

(1) G is a literal, or

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

A disjunction of finite g-literals in LF(X) is called a g-clause.

More detailed notations, concepts and results about α-resolution principle in LP(X) and LF(X) can be found in [28,29,30].

2.2. *α-Generalized resolution in lattice-valued logic based on LIA*

Definition 10. [34] Let $g_1, g_2, \cdots, g_n$ be generalized literals in LP(X). A logical formula Φ is called a general generalized clause if these generalized literals are connected by $\wedge, \vee, \to, '$ and $\leftrightarrow$, denoted by $\Phi(g_1, g_2, \cdots, g_n)$.

Definition 11. [34] A general generalized clause G in LP(X) is called a constant clause if only constants exist in G. Particularly, if for any valuation γ, such that $\gamma(G) = \alpha$, then G is called an α-constant clause.

Definition 12. [34] Let Φ be a general generalized clause in LP(X). A generalized literal g of Φ is called a local extremely complex form, if

(1) g can not be expanded to a more complex generalized literal in Φ by adding $\to$ and $'$.

(2) If g is connected by $\leftrightarrow$, then g is the local extremely complex form as a whole.

All the generalized literals mentioned in this paper are the local extremely complex forms in their corresponding general generalized clauses.

Definition 13. [34] Let $\Phi_1, \Phi_2, \cdots, \Phi_n$ be general generalized clauses in LP(X), $H_i (i = 1, 2, \cdots, n)$ the set of generalized literals in Φ_i, respectively. $\alpha \in L$. If there exists $g_i \in H_i$, such that $\wedge_{i=1}^{n} g_i \leqslant \alpha$, then

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

is called an α-generalized resolvent of $\Phi_1, \Phi_2, \cdots, \Phi_n$, denoted by $G = R_{(g-\alpha)-g}(\Phi_1(g_1), \Phi_2(g_2), \cdots, \Phi_n(g_n))$.

Definition 14. [34] Suppose S is a set of general generalized clauses in LP(X), $\alpha \in L$. Then $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized resolution deduction from S to the general generalized clause D_m, if

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

(2) There exist $r_1, r_2, \cdots, r_k < i$, such that $R_{(g-\alpha)-g}(D_{r_1}, D_{r_2}, \cdots, D_{r_k}) = D_i$.

If there exists an α-generalized resolution deduction from S to an α-constant clause, then w is called an α-generalized refutation.

Theorem 3. [34] *Let S be a set of general generalized clauses in LP(X), $\alpha \in L$, $\{D_1, D_2, \cdots, D_m\}$ an α-generalized resolution deduction from S to the general generalized clause D_m. If $D_m = \alpha$, then $S \leqslant \alpha$.*

Theorem 4. [34] *Let S be a set of general generalized clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ in LP(X), H_i the set of generalized literals in $\Phi_i (i = 1, 2, \cdots, n)$. Suppose there exist quasi-normal generalized literals $g_i \in H_i$, such that $\wedge_{i=1}^{n} g_i \leqslant \alpha$, g_i and S^* are independent of each other if $g_i \notin S^*$, where $S^* \subseteq S$. If $S \leqslant \alpha$, then there exists an α-generalized refutation of S.*

Definition 15. [34] Let $g_1, g_2, \cdots, g_n$ be g-literals, a logical formula in LF(X) is called a general g-clause if these g-literals are connected by $\wedge, \vee, \rightarrow, '$ and $\leftrightarrow$, denoted by $\Phi(g_1, g_2, \cdots, g_n)$.

The general generalized clause in LP(X) is the ground form of general g-clause in LF(X).

Definition 16. [34] Let Φ be a general g-clause in LF(X). If there exists a most general unifier σ of g-literals $g_1, g_2, \cdots, g_m$ in Φ, then Φ^{σ} is called a factor of Φ.

Definition 17. [34] Let $\Phi_1, \Phi_2, \cdots, \Phi_n$ be general g-clauses in LF(X), $\Phi_1^{\sigma_1}$ a factor of Φ_1 for g-literals $g_{11}, g_{12}, \cdots, g_{1r_1}$, $\Phi_2^{\sigma_2}$ a factor of Φ_2 for g-literals $g_{21}, g_{22}, \cdots, g_{2r_2}$, $\cdots$, and $\Phi_n^{\sigma_n}$ a factor of Φ_n for g-literals $g_{n1}, g_{n2}, \cdots, g_{nr_n}$, $\alpha \in L$. If $\wedge_{i=1}^{n} g_{i1}^{\sigma_i} \leqslant \alpha$, then

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

is called an α-generalized resolvent of $\Phi_1, \Phi_2, \cdots, \Phi_n$, denoted by $G = R_{(g-\alpha)-g}(\Phi_1, \Phi_2, \cdots, \Phi_n)$.

Definition 18. [34] Suppose S is a set of general g-clauses in LF(X), $\alpha \in L$. $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized resolution deduction from S to the general g-clause D_m, if

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

(2) There exist $r_1, r_2, \cdots, r_k < i$, such that $R_{(g-\alpha)-g}(D_{r_1}, D_{r_2}, \cdots, D_{r_k}) = D_i$.

Theorem 5. [34] *Let S be a set of general g-clauses in LF(X), $\alpha \in L$, $\{D_1, D_2, \cdots, D_m\}$ an α-generalized resolution deduction from S to the general g-clause D_m. If $D_m = \alpha$, then $S \leqslant \alpha$.*

Theorem 6. [34] *Let S be a set of general g-clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ in LF(X), H_i the set of g-literals in $\Phi_i (i = 1, 2, \cdots, n)$. Suppose there exist quasi-normal g-literals $g_i \in H_i$, such that $\wedge_{i=1}^{n} g_i \leqslant \alpha$, g_i and S^* are independent of each other if $g_i \in S^*$, where $S^* \subseteq S$. If $S \leqslant \alpha$, then there exists an α-generalized refutation of S.*

3. α-Generalized lock resolution in lattice-valued logic

3.1. α-Generalized lock resolution in LP(X)

Definition 19. Let Φ be a general generalized clause in LP(X). Φ is said to be locked if and only if for each generalized literal g in Φ, there exists a positive integer i such that i is the index of g. This specific general generalized clause Φ is called a locked general generalized clause.

Definition 20. Let $\Phi_1, \Phi_2, \cdots, \Phi_n$ be locked general generalized clauses in LP(X), H_i the set of generalized literals in Φ_i, $\alpha \in L$. If there exists g_i with the minimal index in $H_i (i = 1, 2, \cdots, n)$, such that $\wedge_{i=1}^{n} g_i \leqslant \alpha$, then

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

is called an α-generalized lock resolvent of Φ_1, Φ_2, $\cdots, \Phi_n$, denoted by $\Phi = R_{\alpha-g-L}(\Phi_1(g_1), \Phi_2(g_2), \cdots, \Phi_n(g_n))$.

α-Generalized lock resolution is α-lock resolution if the general generalized clause set is taken by its conjunctive normal form.

Definition 21. Suppose S is a set of locked general generalized clauses in LP(X), $\alpha \in L$. Then $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized lock resolution deduction from S to the general generalized clause D_m, if

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

(2) There exist $r_1, r_2, \cdots, r_k < i$, such that $R_{\alpha-g-L}(D_{r_1}, D_{r_2}, \cdots, D_{r_k}) = D_i$.

Remark 3.

(1) In α-generalized lock resolution, the indices of the generalized literals in resolvents are the same with those in their parents. If the resolution level α is generated by substitution in α-generalized lock resolution, then α does not resolve in any next step. Hence, the new generated constant α does not inherit the lock index of resolved literal g_i, and has no index. Moreover, if α is generated as a unit generalized clause, then α can be deleted which does not affect its α-unsatisfiability.

(2) In α-generalized lock resolution deduction, bath resolution can guarantee its completeness and improve the efficiency of α-lock resolution. Sometimes, only resolving on two generalized literals may not derive an α-generalized lock refutation for some α-unsatisfiable formulae. A simple counterexample is: Let $L_9 = \{a_i | 1 \leqslant i \leqslant 9\}$ be a Łukasiewicz implication algebra, x, y propositional variables in L_9P(X), $S = \{x, x \rightarrow y, y \rightarrow a_2\}$. If we take resolution level $\alpha = a_6$, then $S = x \wedge (x \rightarrow y) \wedge (y \rightarrow a_2) \leqslant a_6$, i.e., one resolution step can lead to α-generalized lock refutation. However, any two generalized literals in S are not α-resolved, hence we can not get an α-generalized lock refutation if the number of resolved literals is limited to 2.

Theorem 7. *(Soundness of ground α-generalized lock resolution) Suppose S is a set of locked general generalized clauses in LP(X), $\{D_1, D_2, \cdots, D_m\}$ is an α-generalized lock resolution deduction from S to the general generalized clauses D_m. If $D_m = \alpha$, then $S \leqslant \alpha$.*

Proof. *It follows directly by Theorem 3.* □

Proposition 8. *Let S be a set of locked general generalized clauses $S = \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$ in LP(X), and g_1 a generalized literal of Φ_1. If $S \leqslant \alpha$, then $S_1 = \{\Phi_1(g_1 = \alpha) | \Phi_1 \in S\} \leqslant \alpha$.*

Proof. *We only convert Φ_1 into its conjunctive normal form, i.e., $\Phi_1 = G_1 \wedge G_2 \wedge \cdots \wedge G_m$, where G_i is a generalized clause, and $G_i = g_{i_1} \vee g_{i_2} \vee \cdots \vee g_{i_m}$, g_{i_j} is a generalized literal $(1 \leqslant i \leqslant m; 1 \leqslant j \leqslant m)$. Without loss of generality, suppose only the generalized clause G_1 includes g_1, we denote $S = ((g_1 \vee G_1^0) \wedge G_2 \wedge \cdots \wedge G_m) \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$, where G_1^0 is the disjunction of generalized literals in G_1 except for g_1. Hence, $S = (g_1 \vee G_1^0) \wedge G_2 \wedge \cdots \wedge G_m \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$. Let $S_0 = G_2 \wedge \cdots \wedge G_m \wedge \Phi_2 \wedge \cdots \wedge \Phi_n$, then $S = (g_1 \vee G_1^0) \wedge S_0 = (g_1 \wedge S_0) \vee (G_1^0 \wedge S_0)$. Since $S \leqslant \alpha$, then $G_1^0 \wedge S_0 \leqslant \alpha$. Therefore, $S_1 = \{\Phi(g_1 = \alpha) | \Phi \in S\} = (\alpha \wedge S_0) \vee (G_1^0 \wedge S_0) \leqslant \alpha \vee (G_1^0 \wedge S_0) \leqslant \alpha$.* □

Theorem 9. *(Weak completeness of ground α-generalized lock resolution) Suppose S is the set of locked general generalized clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ in LP(X), where the same generalized literals have the same indices. If $S \leqslant \alpha$, then there exists an α-generalized lock refutation of S.*

Proof. *Let $K(S)$ denote the differences of the number of generalized literals which are locked minus the number of general generalized clauses in S.*

If $K(S) = 0$, then all the locked general generalized clauses of S are unit, in this case, the indices *play no roles in α-generalized lock resolution deduction. Hence, by Theorem 4, S has an α-generalized resolution refutation, which is just an α-generalized lock refutation.*

Suppose that Theorem 9 is true for $K(S) < n$, then there exists at least a general generalized clause in S which is not locked and unit. Since $S \leqslant \alpha$, then there exist generalized literals $g_1, g_2, \cdots, g_m (m \leqslant n)$ such that $g_1 \wedge g_2 \wedge \cdots \wedge g_m \leqslant \alpha$, where $g_i \in \Phi_i (1 \leqslant i \leqslant m)$. Let g_i be a generalized literal with the largest index in Φ_i, denote $S_1 = \{\Phi_1(g_1 = \alpha) | \Phi_1 \in S\}$. By Proposition 8, we have $S_1 \leqslant \alpha$. Since the new substituted constant α has no index, then $K(S_1) < n$. By the induction hypothesis, there exists an α-generalized lock refutation D_1^0 of S_1.

Now we renew D_1^0 by adding g_1 to $\Phi_1(g_1 = \alpha)$ where g_1 is substituted by α in Φ_1, then we get a new α-generalized lock resolution deduction D_1 from S to a general generalized clause Φ_1^0. If all the generalized literals in Φ_1 are not involved in D_1^0, then Φ_1^0 is an α-constant clause. Hence D_1 is also an α-generalized lock refutation of S, i.e., Theorem 9 holds. Otherwise, g_1 has the largest index in Φ_1, and the same generalized literals have the same indices, hence g_1 does not resolve on in D_1. Therefore, Φ_1^0 only includes the generalized literal g_1, and $\Phi_1^0(g_1 = \alpha)$ is an α-constant clause.

Denote $S_2 = \{\Phi_2(g_2 = \alpha) | \Phi_2 \in S\}$. By Proposition 8, we have $S_2 \leqslant \alpha$. Obviously, $K(S_2) < n$. By the induction hypothesis, there exists an α-generalized lock refutation D_2^0 of S_2. We renew D_2^0 to D_2 by adding g_2 to $\Phi_2(g_2 = \alpha)$, where g_2 is substituted by α in Φ_2, then D_2 is an α-generalized lock resolution deduction from S to a general generalized clause Φ_2^0. For Φ_2^0, two cases follow. If Φ_2^0 is an α-constant clause, then D_2 is also an α-generalized lock refutation of S, i.e., Theorem 9 holds. Otherwise, Φ_2^0 is a general generalized clause, which only includes g_2, and $\Phi_2^0(g_2 = \alpha)$ is an α-constant clause.

Repeating the above steps from S_3 to S_m, we denote $S_m = \{\Phi_m(g_m = \alpha) | \Phi_m \in S\}$. Similarly, we have $S_m \leqslant \alpha$ and $K(S_m) < n$. By the induction hypothesis, there exists an α-generalized lock refutation D_m^0 of S_m. We renew D_m^0 to D_m by adding g_m to $\Phi_m(g_m = \alpha)$, where g_m is substituted by α in Φ_m, then D_m is an α-generalized lock resolution deduction from S to a general generalized clause Φ_m^0. If Φ_m^0 is an α-constant clause, then D_m is also an α-generalized lock refutation of S, i.e., Theorem 9 holds. Otherwise, Φ_2^0 is a general generalized clause, which only includes g_m, then $\Phi_m^0(g_m = \alpha)$ is an α-constant clause. Since $\Phi_i^0 (1 \leqslant i \leqslant m)$ only includes g_i, then g_i has the minimal index of Φ_i^0. Furthermore, from $g_1 \wedge g_2 \wedge \cdots \wedge g_m \leqslant \alpha$, it follows that $\vee_{i=1}^m \Phi_i^0(g_i = \alpha) = \alpha$ is an α-generalized lock resolvent of Φ_1^0, Φ_2^0, $\cdots, \Phi_m^0$. Therefore, we connect the α-generalized lock resolution branches $D_1, D_2, \cdots, D_m$, and denote $D = D_1 \cup D_2 \cup \cdots \cup D_m$, then D is an α-generalized lock refutation of S. □

Example 1. Let $L_9 = \{a_i | 1 \leqslant i \leqslant 9\}$ be a Łukasiewicz implication algebra, x, y propositional variables in L_9P(X), $S = \{(x \to a_7) \wedge (x \to y), x \wedge ((x \to y)' \vee ((x \to y) \to a_2)), (z \to a_7) \vee (x \to (x \to a_7)), (w \to z)', x \to y\}$. If we take $\alpha = a_6$, then $S \leqslant a_6$. Then we assign the indices to each generalized literal in S, and get an α-generalized lock refutation of S as follows.

(1) ${}_1(x \to a_7) \wedge {}_2(x \to y)$

(2) ${}_3x \wedge ({}_4(x \to y)' \vee {}_5((x \to y) \to a_2))$

(3) ${}_6(z \to a_7) \vee {}_7(x \to (x \to a_7))$

(4) ${}_8(w \to z)'$

(5) ${}_2(x \to y)$

---

(6) $(a_6 \wedge {}_2(x \to y)) \vee (a_6 \wedge ({}_4(x \to y)' \vee {}_5((x \to y) \to a_2)))$

(7) $a_6 \vee {}_7(x \to (x \to a_7))$

---

(8) $a_6 \wedge ({}_4(x \to y)' \vee {}_5((x \to y) \to a_2))$

---

(9) $a_6 \wedge {}_2(x \to y) \vee {}_5((x \to y) \to a_2)$

(10) $a_6 \wedge {}_5((x \to y) \to a_2)$

---

(11) $a_6 \vee {}_5((x \to y) \to a_2)$

(12) a_6

However, 34 generalized clauses are generated for α-generalized resolution. In this case, α-generalized lock resolution improves the efficiency of α-generalized resolution.

3.2. α-Generalized lock resolution in LF(X)

Definition 22. Let Φ be a general g-clause in LF(X). Φ is said to be locked if and only if for each g-literal g in Φ, there exists a positive integer i such that i is the index of g. This specific general g-clause Φ is called a locked general g-clause.

The locked general g-clause in LF(X) is the general form of locked general generalized clause in LP(X).

Definition 23. Let Φ be a locked general g-clause in LF(X). If there exists a most general unifier σ of g-literals g_1, g_2, $\cdots$, g_m in Φ, then Φ^{σ} is called a locked factor of Φ.

Definition 24. Let $\Phi_1, \Phi_2, \cdots, \Phi_n$ be locked general g-clauses in LF(X), $\Phi_1^{\sigma_1}$ a factor of Φ_1 for g-literals $g_{11}, g_{12}, \cdots, g_{1r_1}$, $\Phi_2^{\sigma_2}$ a factor of Φ_2 for g-literals $g_{21}, g_{22}, \cdots, g_{2r_2}$, $\cdots$, and $\Phi_n^{\sigma_n}$ a factor of Φ_n for g-literals $g_{n1}, g_{n2}, \cdots, g_{nr_n}$, $\alpha \in L$. If there exists g_{i1} with the minimal index in $\Phi_i^{\sigma_i} (i = 1, 2, \cdots, n)$, such that $\wedge_{i=1}^{n} g_{i1} \leqslant \alpha$, then

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

is called an α-generalized lock resolvent of Φ_1, Φ_2, $\cdots, \Phi_n$, denoted by $\Phi = R_{\alpha-g-L}(\Phi_1(g_{11}), \Phi_2(g_{21}), \cdots, \Phi_n(g_{n1}))$.

Definition 25. Suppose S is a set of locked general g-clauses in LF(X), $\alpha \in L$. Then $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized lock resolution deduction from S to the general g-clause D_m, if

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

(2) There exist $r_1, r_2, \cdots, r_k < i$, such that $R_{\alpha-g-L}(D_{r_1}, D_{r_2}, \cdots, D_{r_k}) = D_i$.

Note that Lifting Lemma for α-lock resolution [6] does not rely on the structure of the generalized clauses, hence it still holds for generalized form although the number of resolved literals is extended from 2 to n.

Theorem 10. *Suppose $\Phi_1, \Phi_2, \cdots, \Phi_n$ are locked general g-clauses in LF(X), $\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0$ are instances of general g-clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively. Let τ be a substitution, the index of g-literal g^{τ} in Φ_i^{τ} be the index of g in $\Phi_i (i = 1, 2, \cdots, n)$. If P^0 is an α-generalized lock resolvent of Φ_1^0, $\Phi_2^0, \cdots, \Phi_n^0$, then there exists an α-generalized lock resolvent P of $\Phi_1, \Phi_2, \cdots, \Phi_n$, and a substitution λ such that $P^{\lambda} = P^0$.*

Proof. *Since any general g-clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ can always become two general g-clauses without common variables by a substitution of rename, then we assume that $\Phi_1, \Phi_2, \cdots, \Phi_n$ have no common variables.*

Since $\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0$ are the instances of locked general g-clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively, then there exists a substitution ε, such that $\Phi_1^0 = \Phi_1^{\varepsilon}$, $\Phi_2^0 = \Phi_2^{\varepsilon}, \cdots, \Phi_n^0 = \Phi_n^{\varepsilon}$, and all the generalized literals in Φ_i^{ε} have the same indices with those in $\Phi_i (i = 1, 2, \cdots, n)$. Let $(\Phi_1^0)^{\sigma_1}, (\Phi_2^0)^{\sigma_2}, \cdots, (\Phi_n^0)^{\sigma_n}$ be locked factors of $\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0$, respectively, where σ_1 is a most general unifier g_{11}^{ε}, g_{12}^{ε}, $\cdots$, g_{1r}^{ε} in Φ_1^0, σ_2 is a most general unifier g_{21}^{ε}, g_{22}^{ε}, $\cdots$, g_{2r}^{ε} in Φ_2^0, $\cdots$, σ_n is a most general unifier g_{n1}^{ε}, g_{n2}^{ε}, $\cdots$, g_{nr}^{ε} in Φ_n^0 with g_{i1}^{ε} having the minimal index in $\Phi_i^0 (i = 1, 2, \cdots, n)$. Hence, $(g_{11}^{\varepsilon})^{\sigma_1} = (g_{12}^{\varepsilon})^{\sigma_1} = \cdots = (g_{1r}^{\varepsilon})^{\sigma_1}$, $(g_{21}^{\varepsilon})^{\sigma_2} = (g_{22}^{\varepsilon})^{\sigma_2} = \cdots = (g_{2r}^{\varepsilon})^{\sigma_2}$, $\cdots$, $(g_{n1}^{\varepsilon})^{\sigma_n} = (g_{n2}^{\varepsilon})^{\sigma_n} = \cdots = (g_{nr}^{\varepsilon})^{\sigma_n}$. If $\wedge_{i=1}^{n} (g_{i1}^{\varepsilon})^{\sigma_i} \leqslant \alpha$, then $P^0 = R_{\alpha-g-L}(\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0) = \vee_{i=1}^{n} \Phi_i^{\varepsilon \cdot \sigma_i} (g_{i1}^{\varepsilon \cdot \sigma_i} = \alpha)$.

Since $\varepsilon \cdot \sigma_i$ is a unifier of $g_{i1}, g_{i2}, \cdots, g_{ir}$, then there exists a most general unifier λ_i, such that $g_{i1}^{\lambda_i} = g_{i2}^{\lambda_i} = \cdots = g_{ir}^{\lambda_i}$. Hence, there exists a substitution β_i such that $\varepsilon \cdot \sigma_i = \lambda_i \cdot \beta_i$, where $i = 1, 2, \cdots, n$. From the hypothesis that $\Phi_1, \Phi_2, \cdots, \Phi_n$ have no common variables, it follows that $\lambda_1, \lambda_2, \cdots, \lambda_n$ have no common variables, and $\beta_1, \beta_2, \cdots, \beta_n$ have no common variables. Let $\lambda = \lambda_1 \cup \lambda_2 \cup \cdots \cup \lambda_n$, $\beta = \beta_1 \cup \beta_2 \cup \cdots \cup \beta_n$, by the properties of substitution, $\lambda \cdot \beta = (\lambda_1 \cdot \beta_1)(\lambda_2 \cdot \beta_2) \cup \cdots \cup (\lambda_n \cdot \beta_n)$. Hence, for any $i \in \{1, 2, \cdots, n\}$, $g_{i1}^{\lambda \cdot \beta} = g_{i2}^{\lambda \cdot \beta} = \cdots = g_{ir}^{\lambda \cdot \beta}$. In this case, $\wedge_{i=1}^{n} g_{i1}^{\lambda} \leqslant \alpha$. Furthermore, all the generalized liter-

als in Φ_i^{ε} have the same indices with those in $\Phi_i(i = 1,2,\cdots,n)$, hence g_{i1} has the minimal index in $\Phi_i(i = 1,2,\cdots,n)$. Therefore, $P = R_{\alpha-g-L}(\Phi_1,\Phi_2,\cdots,\Phi_n) = \vee_{i=1}^{n}\Phi_i^{\lambda}(g_{i1}^{\lambda} = \alpha)$.

Moreover,

$$\begin{aligned} P^0 &= R_{\alpha-g-L}(\Phi_1^0,\Phi_2^0,\cdots,\Phi_n^0) \\ &= \vee_{i=1}^{n}\Phi_i^{\varepsilon\cdot\sigma_i}(g_{i1}^{\varepsilon\cdot\sigma_i} = \alpha) \\ &= \vee_{i=1}^{n}\Phi_i^{\lambda\cdot\beta}(g_{i1}^{\lambda\cdot\beta} = \alpha) \\ &= (\vee_{i=1}^{n}\Phi_i^{\lambda}(g_{i1}^{\lambda} = \alpha))^{\beta} \\ &= (P)^{\beta} \end{aligned}$$

□

Theorem 11. *(Soundness of α-generalized lock resolution) Let S be a set of locked general g-clauses in LF(X), $\alpha \in L$, $\{D_1,D_2,\cdots,D_m\}$ an α-generalized lock resolution deduction from S to the general g-clause D_m. If $D_m = \alpha$, then $S \leqslant \alpha$.*

Proof. *It follows directly by Theorem 5.* □

Theorem 12. *(Weak completeness of α-generalized lock resolution) Suppose S is a set of locked general g-clauses in LF(X), where the same g-literals have the same indices. If $S \leqslant \alpha$, then there exists an α-generalized lock refutation of S.*

Proof. *By Theorem 4.5 in [34] and $S \leqslant \alpha$, there exists a finite ground instances set S^0 of S such that $S^0 \leqslant \alpha$. By Theorem 9, there exists a ground α-generalized lock refutation of S^0. From Lifting Lemma of α-generalized lock resolution (Theorem 10), there exists an α-generalized lock refutation of S.* □

4. Equivalent transformation of α-generalized lock resolution

Theorem 13. *Suppose S is a set of locked general g-clauses in LF(X), $\alpha \in L$. There exists an α-generalized lock resolution deduction $w = \{D_1,D_2,\cdots,D_m\}$ from S to D_m in LF(X) if and only if there exists an α-generalized lock resolution deduction $w^0 = \{D_1^0,D_2^0,\cdots,D_m^0\}$ from S^0 to D_m^0 in LP(X), where each of the locked generalized literals in S^0 and D_i^0 are the ground instances of locked g-literals in S and $D_i(1 \leqslant i \leqslant m)$, respectively.*

Proof. *(Necessity) All of the locked generalized literals of S^0 are the ground instances of corresponding locked g-literals in S. For each α-generalized lock resolvent $D_i(1 \leqslant i \leqslant m)$ in LF(X), there exist locked general g-clauses $\Phi_1,\Phi_2,\cdots,\Phi_n$, such that $D_i = R_{\alpha-g-L}(\Phi_1,\Phi_2,\cdots,\Phi_n)$, where $g_1,g_2,\cdots,g_n$ are the α-generalized lock resolved g-literals in $\Phi_1,\Phi_2,\cdots,\Phi_n$, respectively. Hence, $g_1 \wedge g_2 \wedge\cdots\wedge g_n \leqslant \alpha$, and g_i has the minimal index in $\Phi_i(1 \leqslant i \leqslant m)$. Since $g_1 \wedge g_2 \wedge\cdots\wedge g_n \leqslant \alpha$, then for any interpretation $I_D =< D,\mu_D,\nu_D >$ in LF(X), such that $\nu_D(g_1 \wedge g_2 \wedge\cdots\wedge g_n) \leqslant \alpha$. Let $g_1^0,g_2^0,\cdots,g_n^0$ be the ground instances of $g_1,g_2,\cdots,g_n$, respectively, for any valuation function ν_H of $g_1^0,g_2^0,\cdots,g_n^0$ in LP(X), there exists a corresponding H-interpretation I_H, such that $\nu_H(g_1^0 \wedge g_2^0 \wedge\cdots\wedge g_n^0) = \nu_H(g_1 \wedge g_2 \wedge\cdots\wedge g_n) \leqslant \alpha$. Moreover, the indices of generalized literals in Φ_i^0 are the same with those of g-literals in Φ_i, hence g_i^0 has the minimal index of $\Phi_i^0(1 \leqslant i \leqslant n)$. Hence, $D_i^0 = R_{\alpha-g-L}(\Phi_1^0,\Phi_2^0,\cdots,\Phi_n^0)$ is an α-generalized lock resolvent of $\Phi_1,\Phi_2,\cdots,\Phi_n,D_i^0$ is the ground instance of D_i.*

(Sufficiency) It can be proved by Theorem 10 (Lifting Lemma of α-generalized lock resolution). □

Theorem 14. *Suppose S is a set of locked general generalized clauses in $\mathscr{L}_{V(n\times 2)}P(X)$, $\alpha = (h_k,t)$. There exists a (h_k,t)-generalized locked resolution deduction $w = \{D_1,D_2,\cdots,D_m\}$ from S to D_m in $\mathscr{L}_{V(n\times 2)}P(X)$ if and only if there exists a (a_k,b_2)-generalized locked resolution deduction $w^* = \{D_1^*,D_2^*,\cdots,D_m^*\}$ from S^* to D_m^* in $L_{n\times 2}P(X)$, where each of the locked generalized literals in S^* and D_i^* denote the restrictions of locked generalized literals in S and $D_i(1 \leqslant i \leqslant m)$ on $L_{n\times 2}P(X)$, respectively.*

Proof. *We only prove the necessity, the sufficiency can be obtained similarly.*

Let ϕ be an isomorphic mapping as ϕ: $\mathscr{L}_{V(n\times 2)} \longrightarrow \mathscr{L}_{n\times 2}$, then ϕ can be expanded to an isomorphic mapping from $\mathscr{L}_{V(n\times 2)}P(X)$ to $L_{n\times 2}P(X)$, denoted by ϕ_1. Hence, for any formula Φ in $\mathscr{L}_{V(n\times 2)}P(X)$, $\Phi^* = \phi_1(\Phi)$, and Φ^* belongs to

$L_{n\times2}P(X)$. Furthermore, for a valuation γ_1 in $\mathscr{L}_{V(n\times2)}P(X)$, construct $\gamma = \phi\cdot\gamma_1\cdot\phi_1^{-1}$, then it is easy to validate γ is a valuation in $L_{n\times2}P(X)$, and $\gamma_1 = \phi^{-1}\cdot\gamma\cdot\phi_1$.

For each (h_k,t)-generalized lock resolvent $D_i(1\leqslant i\leqslant m)$ in $\mathscr{L}_{V(n\times2)}P(X)$, there exist locked general generalized clauses $\Phi_1,\Phi_2,\cdots,\Phi_n$ such that $D_i = R_{\alpha-g-L}(\Phi_1,\Phi_2,\cdots,\Phi_n)$. Let $g_1,g_2,\cdots,g_n$ be (h_k,t)-generalized lock resolved literals in $\Phi_1,\Phi_2,\ldots,\Phi_n$, respectively, then $g_1\wedge g_2\wedge\cdots\wedge g_n\leqslant(h_k,t)$, and g_i has the minimal index in $\Phi_i(1\leqslant i\leqslant n)$. Hence, for any valuation γ_1 in $\mathscr{L}_{V(n\times2)}P(X)$, we have $\gamma_1(g_1\wedge g_2\wedge\cdots\wedge g_n)\leqslant(h_k,t)$, that is, $\phi^{-1}\cdot\gamma\cdot\phi_1(g_1\wedge g_2\wedge\cdots g_n)\leqslant(h_k,t)$. Since ϕ_1 is monotonic increasing in $\mathscr{L}_{V(n\times2)}P(X)$, then $\gamma\cdot\phi_1(g_1\wedge g_2\wedge\cdots\wedge g_n)\leqslant\phi((h_k,t)) = (a_k,b_2)$. So $\gamma(\phi_1(g_1\wedge g_2\wedge\cdots\wedge g_n))\leqslant(a_k,b_2)$, i.e., $\gamma(g_1^*\wedge g_2^*\wedge\cdots\wedge g_n^*)\leqslant(a_k,b_2)$. By the arbitrariness of γ_1 in $\mathscr{L}_{V(n\times2)}P(X)$, γ is arbitrary in $L_{n\times2}P(X)$. Therefore, $g_1^*\wedge g_2^*\wedge\cdots\wedge g_n^*\leqslant(a_k,b_2)$. Furthermore, the indices of g_i^* in Φ_i^* are the same with g_i in Φ_i, hence g_i^* has the minimal index in $\Phi_i^*(1\leqslant i\leqslant n)$. Therefore, $D_i^* = R_{\alpha-g-L}(\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*)$ is an α-generalized lock resolvent of $\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*$, and D_i^* is the restrictions of D_i on $L_{n\times2}P(X)$. □

Corollary 15. *Suppose S is a set of locked general generalized clauses in $\mathscr{L}_{V(n\times2)}P(X)$, $\alpha = (h_k,f)$. There exists a (h_k,f)-generalized lock resolution deduction $w = \{D_1,D_2,\cdots,D_m\}$ from S to D_m in $\mathscr{L}_{V(n\times2)}P(X)$ if and only if there exists a (a_k',b_1)-generalized lock resolution deduction $w^* = \{D_1^*,D_2^*,\cdots,D_m^*\}$ from S^* to D_m^* in $L_{n\times2}P(X)$, where each of locked generalized literals in S^* and D_i^* denote the restrictions of those in S and $D_i(1\leqslant i\leqslant m)$ on $L_{n\times2}P(X)$, respectively.*

Theorem 16. *Suppose S is a set of locked general generalized clauses in $L_{n\times2}P(X)$, $\alpha = (a_k,b_2)$. There exists a (a_k,b_2)-generalized lock resolution deduction $w = \{D_1,D_2,\cdots,D_m\}$ from S to D_m in $L_{n\times2}P(X)$ if and only if there exists an a_k-generalized lock resolution deduction $w^* = \{D_1^*,D_2^*,\cdots,D_m^*\}$ from S^* to D_m^* in $L_nP(X)$, where each of locked generalized literals in S^* and D_i^* denote the restrictions of those in S and $D_i(1\leqslant i\leqslant m)$ on $L_nP(X)$, respectively.*

Proof. *We only prove the necessity, the sufficiency can be obtained similarly.*

For each α-generalized lock resolvent $D_i(1\leqslant i\leqslant m)$ in $L_{n\times2}P(X)$, there exist general generalized clauses $\Phi_1,\Phi_2,\cdots,\Phi_n$, such that $D_i = R_{\alpha-g-L}(\Phi_1,\Phi_2,\cdots,\Phi_n)$. Let $g_1,g_2,\cdots,g_n$ be (a_k,b_2)-generalized lock resolved literals in $\Phi_1,\Phi_2,\cdots,\Phi_n$, respectively, then $g_1\wedge g_2\wedge\cdots\wedge g_n\leqslant(a_k,b_2)$, and $g_1,g_2,\cdots,g_n$ are generalized literals with the minimal indices in $\Phi_1,\Phi_2,\cdots,\Phi_n$, respectively. Hence, for any valuation $\gamma = \gamma_1\times\gamma_2$ in $L_{n\times2}P(X)$, we have $\gamma(g_1\wedge g_2\wedge\cdots\wedge g_n)\leqslant(a_k,b_2)$, where γ_1 and γ_2 are valuations in $L_nP(X)$ and $L_2P(X)$, respectively, so we have $(\gamma_1\times\gamma_2)(g_1\wedge g_2\wedge\cdots\wedge g_n)\leqslant(a_k,b_2)$. Then, we have $(\gamma_1(g_1\wedge g_2\wedge\cdots\wedge g_n),\gamma_2(g_1\wedge g_2\wedge\cdots\wedge g_n))\leqslant(a_k,b_2)$, and $\gamma_1(g_1\wedge g_2\wedge\cdots\wedge g_n)\leqslant a_k$, that is, $\gamma_1(g_1)\wedge\gamma_1(g_2)\wedge\cdots\wedge\gamma_1(g_n)\leqslant a_k$. Denote $g_i^* = \gamma_1(g_i)$, $\Phi_i^* = \gamma_1(\Phi_i)(1\leqslant i\leqslant m)$, i.e., g_i^* and Φ_i^* are the restrictions of g_i and Φ_i on $L_nP(X)$, respectively, then there exist generalized literals $g_1^*,g_2^*,\cdots,g_n^*$ in $\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*$, such that $g_1^*\wedge g_2^*\wedge\cdots\wedge g_m^*\leqslant a_k$. Moreover, Φ_i is syntactically equal to Φ_i^* except for constants, hence all generalized literals of Φ_i have the same indices with generalized literals of Φ_i^*, that is, g_i^* has the minimal index of $\Phi_i^*(1\leqslant i\leqslant m)$. Hence, $D_i^* = R_{\alpha-g-L}(\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*)$ is an α-generalized lock resolvent of $\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*$, and D_i^* is the restriction of D_i on $L_nP(X)$. □

Remark 4.

(1) Similarly equivalent transformations still hold for α-generalized resolution deduction, i.e., α-generalized resolution can be transformed from $\mathscr{L}_{V(n\times2)}$F(X) into $L_{n\times2}$P(X), which can also simplify the complexity of α-generalized resolution.

(2) For α-generalized lock resolution in $\mathscr{L}_{V(n\times2)}$F(X), we can equivalently transform it into that in $\mathscr{L}_{V(n\times2)}$P(X) by Theorem 13, and further to that in $L_{n\times2}$P(X) by Theorem 14, and finally to L_nP(X) by Theorem 16 which takes truth-values in a Łukasiewicz implication algebra on a finite chain L_n, that is, for discussing α-generalized lock resolution in $\mathscr{L}_{V(n\times2)}$F(X), we only need to discuss that in L_nP(X).

5. Compatibilities of α-generalized lock resolution

5.1. α-Generalized linear semi-lock resolution method in $L_nF(X)$

Restricted strategies on resolution methods can reduce the deduction trees, but do not affect their semantic properties. Hence, the soundness of α-generalized linear semi-lock resolution holds, we only need to discuss its completeness.

Definition 26. Suppose S is a set of general g-clauses in LF(X), $\alpha \in L$, Φ_0 is a general g-clause in S. An α-generalized resolution deduction $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized linear resolution deduction from S with the top general g-clause Φ_0 to the general g-clause D_m if it satisfies

(1) D_{i+1} is the α-generalized resolvent of D_i (center g-clause) and B_i (side g-clause), $i = 0, 1, 2, \cdots, m-1$.

(2) For $i \in \{0, 1, 2, \cdots, m-1\}$, $B_i \in S$, or $B_i = D_j$, where $j < i$, $D_0 = \Phi_0$.

Definition 27. Suppose S is a set of locked general g-clauses in LF(X), $\alpha \in L$, Φ_0 is a locked general g-clause in S. An α-generalized linear resolution deduction $w = \{D_1, D_2, \cdots, D_m\}$ is called an α-generalized linear lock resolution deduction from S with the top general g-clause Φ_0 to the general g-clause D_m if it satisfies: For each center g-clause $D_{i+1}(i = 0, 1, 2, \cdots, m-1)$, D_{i+1} is the α-generalized lock resolvent of D_i (center g-clause) and B_i (side g-clause). w is called an α-generalized linear semi-lock resolution deduction from S with the top g-clause Φ_0 to the g-clause D_m if the α-generalized resolved g-literals in $D_i(i = 1, 2, \cdots, m-1)$ have the minimal indices.

Theorem 17. *Suppose S is the set of locked general generalized clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ in $L_nP(X)$, where the same generalized literals have the same indices in $\Phi_i(1 \leqslant i \leqslant n)$, all the indices of generalized literals in Φ_i are less than those in $\Phi_j(i < j)$. If $S \leqslant \alpha$, and $S - \{\Phi_1\}$ is α-satisfiable, then there exists an α-generalized linear semi-lock refutation of S with the top clause Φ_1.*

Proof. *If S only includes a general generalized clause Φ_0, then $\Phi_0 \leqslant \alpha$ by $S \leqslant \alpha$. Hence, Φ_0 has an α-generalized linear semi-lock refutation by resolved on itself.*

Suppose there exist at least two general generalized clauses in S. Let $m(S)$ denote the number of generalized literals which are locked in S. If $m(S) = 2$, then there exist two generalized literals g_1, g_2 in S, i.e., $S = g_1 \wedge g_2$. Obviously, there exists an α-generalized linear semi-lock refutation of S with the top clause g_1.

Suppose Theorem 17 is true for $m(S) < n$. Since $S \leqslant \alpha$, then there exist generalized literals $g_1, g_2, \cdots, g_m$ in $\Phi_1, \Phi_2, \cdots, \Phi_m$, respectively, such that $g_1 \wedge g_2 \wedge \cdots \wedge g_m \leqslant \alpha (m \leqslant n)$. Without loss of generality, let g_i have the largest index of $\Phi_i(1 \leqslant i \leqslant m)$, and S the minimum α-unsatisfiable set which includes Φ_i. Denote $S_1 = \{\Phi_1(g_1 = \alpha) | \Phi_1 \in S\}$, $\Phi_1^0 = \Phi_1(g_1 = \alpha)$. By Proposition 8, we have $S_1 \leqslant \alpha$. Obviously, $m(S_1) < n$. From $S_1 - \{\Phi_1^0\} = S - \{\Phi_1\}$, it follows that $S_1 - \{\Phi_1^0\}$ is α-satisfiable. By the induction hypothesis, there exists an α-generalized linear semi-lock refutation D_1^0 of S_1 with the top clause Φ_1^0. We renew D_1^0 to D_1 by adding g_1 to Φ_1^0, where g_1 is substituted by α, then D_1 is an α-generalized linear semi-lock resolution deduction from S with the top clause Φ_1 to the general generalized clause Ψ_1. For Ψ_1, two cases follow. Ψ_1 is either an α-constant clause, or a general generalized clause which only includes g_1, and $\Psi_1(g_1 = \alpha) = \alpha$.

Similar to S_1, we denote $S_i = \{\Phi_i(g_i = \alpha) | \Phi_i \in S\}$, $\Phi_i^0 = \Phi_i(g_i = \alpha)$, then $S_i \leqslant \alpha$ and $S_i - \{\Phi_i^0\}$ is α-satisfiable $(2 \leqslant i \leqslant m)$. Furthermore, $m(S_i) < n$. By the induction hypothesis, there exists an α-generalized linear semi-lock refutation D_i^0 of S_i with the top clause Φ_i^0. We only renew D_i^0 to D_i by adding g_i to Φ_i^0, where g_i is substituted by α in side clauses of D_i^0, then D_i is an α-generalized linear semi-lock resolution deduction from S with the top clause Φ_i to Ψ_i. For Ψ_i, two cases follow. Ψ_i is either an α-constant clause or a general generalized clause which only includes g_i, and $\Psi_i(g_i = \alpha) = \alpha$.

For the resolvent Ψ_1 in D_1, if Ψ_1 is an α-constant clause, then D_1 is also an α-generalized linear semi-lock refutation of S with the top clause Φ_1, i.e., Theorem 17 holds. Otherwise, Ψ_1 is a general gener-

alized clause, which only includes g_1, *and* $\Psi_1(g_1 = \alpha) = \alpha$. *In this case, resolve on* Ψ_1, $\Phi_2, \cdots, \Phi_m$, *and get* $\Psi_1^* = R_{(g-\alpha)-g}(\Psi_1(g_1), \Phi_2(g_2), \cdots, \Phi_m(g_m)) = \Psi_1(g_1 = \alpha) \vee \Phi_2(g_2 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$, *hence we get an* α*-generalized linear semi-lock deduction* D_1^* *of* S *with the top clause* Φ_1 *to* $\Phi_2(g_2 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$. *Hence, we add* $\Phi_3(g_3 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *to the top clause* $\Phi_2(g_2 = \alpha)$ *in* D_2. *Since the indices of all generalized literals in* Φ_i *are less than those of* $\Phi_j (i < j)$, *then* $\Phi_3(g_3 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *can not be resolved in* D_2. *Hence, we get an* α*-generalized linear semi-lock resolution deduction* D_2^* *of* S *with the top clause* $\Phi_2(g_2 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *to the general generalized clause* $\Psi_2^* = \Psi_2 \vee \Phi_3(g_3 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$.

Similarly, if Ψ_2 *is an* α*-constant clause, then* $\Psi_2^* = \Phi_3(g_3 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$. *Otherwise,* Ψ_2 *is a general generalized clause which only includes* g_2, *and* $\Psi_2(g_2 = \alpha) = \alpha$. *In this case, resolve on* $\Psi_1, \Psi_2, \Phi_2, \cdots, \Phi_m$, *and get* $\Psi_2^* = R_{(g-\alpha)-g}(\Psi_1(g_1), \Psi_2(g_2), \cdots, \Phi_m(g_m)) = \Psi_1(g_1 = \alpha) \vee \Psi_2(g_2 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$, *it is also an* α*-generalized linear semi-lock deduction* D_2^* *of* S *with the top clause* $\Phi_2(g_2 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *to* Ψ_2^*. *Hence, we add* $\Phi_4(g_4 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *to* $\Phi_3(g_3 = \alpha)$ *in* D_3. *Since all the indices of generalized literals in* Φ_i *are less than those of* $\Phi_j (i < j)$, *then* $\Phi_4(g_4 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$ *can not be resolved in* D_3. *Hence, we get an* α*-generalized linear semi-lock resolution deduction* D_3^* *of* S *with the top clause* Ψ_2^* *to the general generalized clause* $\Psi_3^* = \Psi_3 \vee \Phi_4(g_4 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$. *Similarly, if* Ψ_3 *is an* α*-constant clause, then* $\Psi_3^* = \Phi_4(g_4 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$. *Otherwise,* Ψ_3 *is a general generalized clause, which only includes* g_3, *and* $\Psi_3(g_3 = \alpha) = \alpha$. *Hence, we resolve on* $\Psi_1, \Psi_2, \Psi_3, \Phi_4, \cdots, \Phi_m$, *and get* $\Psi_3^* = R_{(g-\alpha)-g}(\Psi_1, \Psi_2, \Psi_3, \Phi_4, \cdots, \Phi_m) = \Psi_1(g_1 = \alpha) \vee \Psi_2(g_2 = \alpha) \vee \Psi_3(g_3 = \alpha) \vee \Phi_4(g_4 = \alpha) \vee \cdots \vee \Phi_m(g_m = \alpha)$, *it is also an* α*-generalized linear semi-lock deduction* D_3^* *of* S *with the top clause* Ψ_2^* *to* Ψ_3^*.

Repeat the similar steps above, we get $\Psi_{m-1}^* = R_{(g-\alpha)-g}(\Psi_1, \Psi_2, \cdots, \Psi_{m-1}, \Phi_m) = \Psi_1(g_1 = \alpha) \vee \Psi_2(g_2 = \alpha) \vee \cdots \vee \Psi_{m-1}(g_{m-1} = \alpha) \vee \Phi_m(g_m = \alpha)$, *it is also an* α*-generalized linear semi-lock deduction* D_{m-1}^* *of* S *with the top clause* Ψ_{m-2}^* *to* Ψ_{m-1}^*. *Since* $\Psi_{m-1}^* = \Phi_m^0$ *and* D_m *is an* α*-generalized linear semi-lock resolution deduction from* S *with the top clause* Φ_m^0 *to* Ψ_m. *If* Ψ_m *is an* α*-constant clause, then connecting the deductions* $D_1^*, D_2^*, \cdots, D_{m-1}^*, D_m$, *we get an* α*-generalized linear semi-lock refutation of* S *with the top clause* Φ_1, *i.e., Theorem 17 holds. Otherwise,* Ψ_m *is a general generalized clause which only includes* g_m, *and* $\Psi_m(g_m = \alpha) = \alpha$. *In this case, we resolve on* $\Psi_1, \Psi_2, \Psi_3, \cdots, \Psi_{m-1}, \Psi_m$, *and get* $\Psi_m^* = R_{(g-\alpha)-g}(\Psi_1, \Psi_2, \Psi_3, \cdots, \Psi_{m-1}, \Psi_m) = \Psi_1(g_1 = \alpha) \vee \Psi_2(g_2 = \alpha) \vee \Psi_3(g_3 = \alpha) \vee \cdots \vee \Psi_{m-1}(g_{m-1} = \alpha) \vee \Psi_m(g_m = \alpha) = \alpha$, *hence we get an* α*-generalized linear semi-lock refutation of* S *with the top clause* Φ_1. *Therefore, we connect the* α*-generalized linear semi-lock resolution branches* $D_1^*, D_2^*, \cdots, D_m^*$, *and denote* $D^* = D_1^* \cup D_2^* \cup \cdots \cup D_m^*$, *then* D^* *is an* α*-generalized linear semi-lock refutation of* S *with the top clause* Φ_1. □

Theorem 18. *Suppose* S *is the set of locked general g-clauses* $\Phi_1, \Phi_2, \cdots, \Phi_n$ *in* $L_nF(X)$, *where the same g-literals have the same indices in* $\Phi_i (1 \leqslant i \leqslant n)$, *all the indices of g-literals in* Φ_i *are less than those of* $\Phi_j (i < j)$. *If* $S \leqslant \alpha$, *and* $S - \{\Phi_1\}$ *is* α*-satisfiable, then there exists an* α*-generalized linear semi-lock refutation of* S *with the top clause* Φ_1.

Proof. *Similar to the proof of Theorem 12, the completeness of* α*-generalized linear semi-lock resolution in* $L_nF(X)$ *follows.* □

5.2. An algorithm for α-generalized linear semi-lock resolution in $L_nP(X)$

In α-generalized resolution, the number of resolved literals is dynamic, hence it is vital to give a method to judge whether the given generalized literals are resolved or not.

Definition 28. Let $g_1, g_2, \cdots, g_m$ be generalized literals in LP(X). $g_1, g_2, \cdots, g_m$ are α-minimum resolved in α-generalized resolution if it satisfies

(1) $g_1 \wedge g_2 \wedge \cdots \wedge g_m \leqslant \alpha$.

(2) For any $\{g_{i1}, g_{i2}, \cdots, g_{ik}\} \subset \{g_1, g_2, \cdots, g_m\}$, $g_{i1} \wedge g_{i2} \wedge \cdots \wedge g_{ik} \nleq \alpha$.

Example 2. Let $L_9 = \{a_i | 1 \leqslant i \leqslant 9\}$ be a Łukasiewicz implication algebra, x, y propositional variables in

$L_9P(X)$, $g_1 = x$, $g_2 = x \to y$, $g_3 = y \to a_2$. If we take resolution level $\alpha = a_6$, then $g_1 \wedge g_2 \wedge g_3 \leqslant a_6$. Moreover, for any $i, j \in \{1,2,3\}$, $g_i \wedge g_j \nleqslant a_6$. Hence, g_1, g_2, g_3 are a_6-minimum resolved in a_6-generalized resolution.

Algorithm 1

Step 0. Given generalized literals $g_1, g_2, \cdots, g_m$ in $L_nP(X)$, $\alpha = a_k$.

Step 1. Judge $g_1 \wedge g_2 \wedge \cdots \wedge g_m \leqslant a_k$. If it does not satisfy, then stop, $Res_m(g_1, g_2, \cdots, g_m, a_k) = 0$.

Step 2. $k = 1$.

Step 3. For any k generalized literals $\{g_{i1}, g_{i2}, \cdots, g_{ik}\} \subset \{g_1, g_2, \cdots, g_m\}$, judge $g_{i1} \wedge g_{i2} \wedge \cdots \wedge g_{ik} \leqslant a_k$, if it satisfies, then stop, $Res_m(g_1, g_2, \cdots, g_m, a_k) = 0$.

Step 4. If $k \leqslant m-1$, then $k = k+1$, go to Step 3. Otherwise, stop, $Res_m(g_1, g_2, \cdots, g_m, a_k) = 1$.

Algorithm 1 can be seen as a function, i.e., Function $Res_m(g_1, g_2, \cdots, g_m, a_k)$. Let $g_1, g_2, \cdots, g_m$ be generalized literals in $L_nP(X)$. $g_1, g_2, \cdots, g_m$ are a_k-minimum resolved if $Res_m(g_1, g_2, \cdots, g_m, a_k) = 1$. Otherwise, it returns 0.

According to Algorithm 1, an algorithm for α-generalized linear semi-lock resolution follows.

Algorithm 2

Step 0. (Initiation) Let S be a set of locked general generalized clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$. Assign to each occurrence of generalized literal a positive integer in Φ_i, the same generalized literals have the same indices, and all the indices of generalized literals in Φ_i are less than those of $\Phi_j (i < j)$. $c = 1, \alpha = a_k$. $S - \{\Phi_1\}$ be a_k-satisfiable. $\Phi = \Phi_1$.

Step 1. Let g be the generalized literal with the minimal index in Φ, H the set of generalized literals in $S - \{\Phi\}$.

Step 2. Let n_0 be the number of generalized literals in H. For $i_0 = 1$ to n_0, If there exist i_0 generalized literals $g_{j1}, g_{j2}, \cdots, g_{ji_0}$ ($g_{ji} \in \Phi_j$), such that $Res_m(g, g_{j1}, g_{j2}, \cdots, g_{ji_0}, a_k) = 1$, then $\Phi_m = R_{\alpha-g-L}(\Phi, \Phi_{j1}, \Phi_{j2}, \cdots, \Phi_{ji_0})$. If $\Phi_m = a_k$, then stop, $S \leqslant a_k$. Otherwise, stop, S is a_k-satisfiable.

Step 3. $S = S \cup \Phi_m$, $c = c+1$. Set $\Phi = \Phi_m$.

Step 4. If $c \leqslant c_0$, then go to Step 1. Otherwise, stop, S is a_k-satisfiable.

Remark 5. c_0 can be chosen according to the complexity of Algorithm 2 and the numbers of generalized literals in S.

Theorem 19. *(Soundness) If Algorithm 2 terminates, then the a_k-unsatisfiability of S can be judged in $L_nP(X)$.*

Proof. *By Algorithm 2, if it terminates, then two cases follow. One is in Step 2, if there exist some a_k-generalized lock resolvent Φ_m, such that $\Phi_m = a_k$, then by the soundness of a_k-generalized linear semi-lock resolution in $L_nP(X)$, we have $S \leqslant a_k$. Otherwise, if no a_k-generalized lock resolvents exists, then S is a_k-satisfiable. Another is in Step 5, if $c > c_0$, then S can be seen to be a_k-satisfiable. Therefore, the a_k-unsatisfiability of logical formulae can be judged if it terminates.* □

Theorem 20. *(Completeness) If $S \leqslant a_k$ in $L_nP(X)$, then Algorithm 2 terminates in Step 2.*

Proof. *If $S \leqslant a_k$, then by Algorithm 2 and the completeness of a_k-generalized linear semi-lock resolution in $L_nP(X)$, there exists an a_k-generalized linear semi-lock refutation of S. Hence, an a_k-constant clause generates in this deduction, that is, Algorithm 2 can terminate in Step 2.* □

6. Conclusion

In this paper, we presented refined non-clausal resolution methods in linguistic truth-valued lattice-valued first-order logic ($\mathscr{L}_{V(n\times 2)}$F(X)), i.e., α-generalized lock resolution and α-generalized linear semi-lock resolution including their concepts, soundness and completeness. We equivalently transformed α-generalized lock resolution in $\mathscr{L}_{V(n\times 2)}$F(X)) to that in $L_nP(X)$, it greatly simplifies its complexity for $\mathscr{L}_{V(n\times 2)}$F(X). In order to further improve the efficiency of α-generalized lock resolution, we discussed α-generalized linear semi-lock resolution by combining it with α-generalized linear resolution. The further research will be concentrated on discussing how to assign appropriate indices to the generalized literals in α-generalized lock resolution, and exploring some theoretical or practical applications for α-generalized resolution based automated reasoning.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant No. 60875034, 61175055) and Sichuan Key Technology Research and Development Program under Grant No.2011FZ0051.

References

1. G. Aguilera, IP De Guzman, M. Ojeda-Aciego and A. Valverde, "Reductions for non-clausal theorem proving," *Theoretical Computer Science*, **266**(1-2): 81–112 (2001).
2. C. Areces and D. Gorín, "Resolution with order and selection for hybrid logics," *Journal of Automated Reasoning*, **46**(1): 1–42 (2011).
3. R. Boyer, "Locking: A restriction of resolution," PhD thesis, University of Texas at Austin (1971).
4. C. Dixon, M. Fisher, M. Wooldridge, "Resolution for temporal logics of knowledge," *Journal of Logic and Computation*, **8**(3): 345–372 (1998).
5. X.X. He, J. Liu, Y. Xu, L. Martínez, D. Ruan, "On α-satisfiability and its α-lock resolution in a finite lattice-valued propositional logic," *Logic Journal of IGPL*, **20**(3): 579–588 (2012).
6. X.X. He, Y. Xu, J. Liu, D. Ruan, "α-Lock resolution method for a lattice-valued first-order logic," *Engineering Applications of Artificial Intelligence*, **24**(7): 1274–1280 (2011).
7. 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*, **16**(4): 699–709 (2012).
8. U. Hustadt, R.A. Schmidt, "Using resolution for testing modal satisfiability and building models," *Journal of Automated Reasoning*, **28**(2): 205–232 (2002).
9. J. Liu, D. Ruan, Y. Xu, Z.M. Song. "A resolution-like strategy based on a lattice-valued logic," *IEEE Transaction on Fuzzy System*, **11**(4): 560–567 (2003).
10. X.H. Liu, "Automated reasoning based on resolution methods," Science Press, Beijing (In Chinese)(1994).
11. X.H. Liu, J.G. Sun, "Generalized resolution and NC-resolution," *Journal of Computer Science and Technology*, **9**(2): 160–167 (1994).
12. J. Ma, W.J. Li, D. Ruan, and Y. Xu. "Filter-based resolution principle for lattice-valued propositional logic LP(X)," *Information Sciences*, **177**(4): 1046–1062 (2007).
13. N.V. Murray. "Completely non-clausal theorem proving," *Artificial Intelligence*, **18**(1): 67–85 (1982).
14. J.A. Navarro, A. Voronkov, "Generation of hard non-clausal random satisfiability problems," *Proceedings of the national conference on artificial intelligence (AAAI)*, **20**(1): 436–442 (2005).
15. P.W. O′Hearn, Z. Stachniak, "A resolution framework for finitely-valued first-order logics," *Journal of Symbolic Computation*, **13**(3): 235–254 (1992).
16. Z. Pei, D. Ruan, J. Liu, Y. Xu, "Linguistic values-based intelligent information processing: theory, methods, and applications," Atlantis Press, Paris (2009).
17. D.A. Plaisted, S. Greenbaum, "A structure-preserving clause form translation," *Journal of Symbolic Computation*, **2**(3): 293–304 (1986).
18. K.Y. Qin, Y. Xu. "Lattice-valued proposition logic (II)," *Journal of Southwest Jiaotong University*, **2**(1): 22–27 (1994).
19. J.A. Robinson, "A machine-oriented logic based on the resolution principle," *Journal of the ACM*, **12**(1): 23–41 (1965).
20. R. Sebastiani, "Applying GSAT to non-clausal formulas," *Journal of Artificial Intelligence Research*, **1**(1): 309–314 (1993).
21. Z. Stachniak, "Non-clausal reasoning with definite theories," *Fundamenta Informaticae*, **48**(4): 363–388 (2001).
22. Z. Stachniak, A. Belov, "Weighting strategy for non-clausal resolution," *Journal of Experiment Theory Artificial Intelligence*, **20**(1): 61–78 (2008).
23. J.G. Sun, X.H Liu, "Generalized linear semi-lock resolution," *Chinese Bulletin*, **37**(19): 1812–1814 (1992).
24. X.H. Wang, X.H. Liu, "Generalized resolution," *The Chinese Journal of Computers*, **5**(2): 81–92 (1982).
25. W.T. Xu, Y. Xu, "α-Generalized linear resolution method based on lattice-valued propositional logic LP(X)," *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 26-28 July, Shanghai, 1413–1417 (2011).
26. Y. Xu, "Lattice implication algebras," *Journal of Southwest Jiaotong University*, **89**(1): 20–27 (in Chinese) (1993).
27. Y. Xu, K.Y. Qin, "Lattice-valued propositional logic (I)," *Journal of Southwest Jiaotong University*, **1**(2): 123–128 (1993).

28. Y. Xu, D. Ruan, E.E. Kerre, J. Liu, "α-Resolution principle based on lattice-valued propositional logic LP(X)," *Information Sciences*, **130**(1-4): 195–223 (2000).
29. Y. Xu, D. Ruan, E.E. Kerre, J. Liu, "α-Resolution principle based on first-order lattice-valued logic LF(X)," *Information Sciences*, **132**(1-4): 221–239 (2001).
30. Y. Xu, D. Ruan, K.Y. Qin, J. Liu, "Lattice-valued logic–An alternative approach to treat fuzziness and incomparability," Springer-Verlag, Berlin (2003).
31. Y. Xu, J. Liu, D. Ruan, T.T. Lee. "On the consistency of rule bases based on lattice-valued first-order logic LF(X)," *International Journal of Intelligent Systems*, **21**(4): 399–424 (2006).
32. Y. Xu, W.T. Xu, X.M. Zhong, X.X. He, "α-Generalized resolution principle based on lattice-valued propositional logic system LP(X)," *The 9th International FLINS Conference on Foundations and Applications of Computational Intelligence (FLINS2010)*, 2-4 August, Emei, China, 66–71 (2010).
33. Y. Xu, J. Liu, D. Ruan, X.B. Li, "Determination of α-resolution in lattice-valued first-order logic LF(X)," *Information Sciences*. **181**(10): 1836–1862 (2011).
34. Y. Xu, X.X. He, J. Liu, S.W. Chen, "A general form of α-generalized resolution based on lattice-valued logic," Submitted to *International Journal of Computational Intelligence Systems* (2011).
35. L.A. Zadeh, "The concept of linguistic variable and its application to approximate reasoning," *Information Sciences*, I: **8**, 199–249; II: **8**, 310–357; III: **9**, 43–80 (1975).
36. L.A. Zadeh, J. Kacprzyk, "Computing with words in information / intelligent system: Foundations," Springer-Verlag, Berlin (1999).