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

Xingxing He¹, Yang Xu¹, Jun Liu², Shuwei Chen²

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 $L_{\mathcal{V}_{[0,2]}}(F(X))$ is equivalently transformed into that for propositional logic $L_n(P(X))$, 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 \( \alpha \)-resolution \(^{28,29} \), Xu et al. \(^{30,32,34} \) proposed \( \alpha \)-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 \( \alpha \)-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.

\( \alpha \)-Lock resolution \(^5,6,7 \) is a simple, but effective refinement on \( \alpha \)-resolution, it is an \( \alpha \)-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 \( \alpha \)-generalized resolution, and propose \( \alpha \)-generalized lock resolution method. Compared with \( \alpha \)-lock resolution \(^7 \), \( \alpha \)-generalized lock resolution can validate \( \alpha \)-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 \( \alpha \)-resolved simultaneously, but not \( \alpha \)-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 \( \alpha \)-generalized resolution to some extent. Furthermore, for improving the efficiency of \( \alpha \)-generalized lock resolution, we propose \( \alpha \)-generalized linear semi-lock resolution by combining it with \( \alpha \)-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 \( \mathcal{L}_{V(n\times 2)}(F(X)) \) \(^{16,30} \) 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 \( \alpha \)-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 \( \mathcal{L}_{V(n\times 2)}(F(X)) \), 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 \( \mathcal{L}_{V(n\times 2)} \) 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 \( \mathcal{L}_{V(n\times 2)}(F(X)) \). The other is equivalently or conditionally transforming the resolution methods from \( \mathcal{L}_{V(n\times 2)}(F(X)) \) 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 $\mathcal{L}_{(\forall x \exists x)} F(X)$. With this in mind, this paper transforms $\alpha$-generalized lock resolution in $\mathcal{L}_{(\forall x \exists x)} F(X)$ into that in $L_n P(X)$ whose truth-valued domain is a Łukasiewicz implication algebra on a finite chain $L_n$, and discusses $\alpha$-generalized linear semi-lock resolution in $L_n P(X)$, which can further improve the efficiency of $\alpha$-generalized resolution.

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

2. Preliminaries

2.1. $\alpha$-Resolution principle in lattice-valued logic based on LIA

Definition 1. 26,30 Let $(L, \lor, \land, O, I)$ be a bounded lattice with an order-reversing involution "~", $I$ and $O$ the greatest and the smallest element of $L$, respectively, and $\rightarrow: L \times L \rightarrow L$ a mapping. $\mathcal{L} = (L, \lor, \land, \rightarrow, O, I)$ is called a lattice implication algebra (LIA) if the following conditions hold for any $x, y, z \in L$:

(I1) $\quad x \rightarrow (y \rightarrow z) = (x \rightarrow y) \lor (x \rightarrow z)$,

(I2) $\quad x \rightarrow x = I$,

(I3) $\quad x \rightarrow y = y' \rightarrow x'$,

(I4) $\quad x \rightarrow y = y \rightarrow x = I$ implies $x = y$,

(I5) $\quad (x \rightarrow y) \rightarrow y = (y \rightarrow x) \rightarrow x$,

($L_1$) $\quad (x \lor y) \rightarrow z = (x \rightarrow z) \land (y \rightarrow z)$,

($L_2$) $\quad (x \land y) \rightarrow z = (x \rightarrow z) \lor (y \rightarrow z)$.

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, $L_n = \{a_i | 1 \leq i \leq n\}$ and $a_1 < a_2 < \ldots < a_n$, define for any $a_i, a_j \in L_n$, $a_i \lor a_j = a_{\max(i,j)}$, $a_i \land a_j = a_{\min(i,j)}$, $(a_i)' = a_{n-i+1}$, $a_i \rightarrow a_j = a_{\min(n-i+j,n)}$, then $\mathcal{L}_n = (L_n, \lor, \land, \rightarrow, a_1, a_n)$ is an LIA.

All elements in Łukasiewicz implication algebra are completely comparable.

Proposition 2. 16, 31 Let $L_n = \{a_1, \ldots, a_n\}$, $a_1 < a_2 < \ldots < a_n$, $L_2 = \{b_1, b_2\}$, $b_1 < b_2$. $(L_n, \lor, \land, \rightarrow, a_1, a_n)$ and $(L_2, \lor, \land, \rightarrow, b_1, b_2)$ 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 $(a_i, b_j) \lor (a_k, b_l) = (a_i \lor a_k, b_j \lor b_l), (a_i, b_j) \land (a_k, b_l) = (a_i \land a_k, b_j \land b_l), (a_i, b_j)' = (a_i', b_j'), (a_i, b_j) \rightarrow (a_k, b_l) = (a_i \rightarrow a_k, b_j \rightarrow b_l)$, then $\mathcal{L}_{n \times 2} = (L_n \times L_2, \lor, \land, \rightarrow, (a_1, b_1), (a_n, b_2))$ is an LIA.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{hasse.png}
\caption{Hasse Diagram of $L_n \times L_2$}
\end{figure}

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 $L_{(\forall x \exists x)} = AD_n \times MT$. Define a mapping $g$ as $g : L_{(\forall x \exists x)} \rightarrow L_n \times L_2$.

$$g((h_i, mt)) = \begin{cases} (a_i', b_1), & mt = f, \\ (a_i, b_2), & mt = t. \end{cases}$$
Then $g$ is bijection, denote its inverse mapping as $g^{-1}$. For any $x, y \in L(V(n \times 2))$, define $x \lor y = g^{-1}(g(x) \lor g(y))$, $x \land y = g^{-1}(g(x) \land g(y))$, $x' = g^{-1}(g(x))$, $x \rightarrow y = g^{-1}(g(x) \rightarrow g(y))$, then $\mathcal{L}(n \times 2) = (L(V(n \times 2)), \lor, \land, ' , \rightarrow , (h_n, f), (h_n, t))$ 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 $\mathcal{L}(n \times 2)$ to $\mathcal{L}$. 

**Definition 3.** 28 Let $X$ be a set of propositional variables, $T = L \cup \{', \rightarrow \}$ be a type with $ar(') = 1$, $ar(\rightarrow) = 2$ 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 $\mathcal{L}(n \times 2)$, this specific $LP(X)$, i.e., $\mathcal{L}(n \times 2)P(X)$, is a linguistic truth-valued lattice-valued propositional logic system. Similarly, the truth-valued domain of $L_nP(X)$ is a Łukasiewicz implication algebra $\mathcal{L}_n$.

**Definition 4.** 28 A valuation of $LP(X)$ is a propositional algebra homomorphism $\gamma$: LP(X) $\rightarrow L$. 

**Definition 5.** 28 Let $F$ be a logical formula in $LP(X)$, $\alpha \in L$. If there exists a valuation $\gamma_0$ of $LP(X)$ such that $\gamma_0(F) \supseteq \alpha$, then $F$ is satisfiable by a truth-value level $\alpha$, in short, $\alpha$-satisfiable. If $\gamma(F) \supseteq \alpha$ for every valuation $\gamma$ of $LP(X)$, $F$ is valid by the truth-value level $\alpha$, in short, $\alpha$-valid. If $\gamma(F) \subseteq \alpha$ for every valuation $\gamma$ of $LP(X)$, $F$ is always false by the truth-value level $\alpha$, in short, $\alpha$-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 $\rightarrow$ and $'$ at most.

2. For any $G \in LP(X)$, if $G \in F$ in $LP(X)$, then $G$ is an ESF containing connectives $\rightarrow$ 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 $\mathcal{L}(V(n \times 2))P(X)$, the truth-valued domain of first order logic systems $\mathcal{L}(V(n \times 2))F(X)$ and $L_nF(X)$ are $\mathcal{L}(n \times 2)$ and $\mathcal{L}_n$, 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 $\lor$ or $\land$ 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 $\alpha$-resolution principle in $LP(X)$ and $LF(X)$ can be found in $28, 29, 30$.

### 2.2. $\alpha$-Generalized resolution in lattice-valued logic based on LIA

**Definition 10.** 34 Let $g_1, g_2, \ldots, g_n$ be generalized literals in $LP(X)$. A logical formula $\Phi$ is called a general generalized clause if these generalized literals are connected by $\land, \lor, \rightarrow$, and $\leftrightarrow$, denoted by $\Phi(g_1, g_2, \ldots, g_n)$.

**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 $\gamma$, such that $\gamma(G) = \alpha$, then $G$ is called an $\alpha$-constant clause.

**Definition 12.** 34 Let $\Phi$ be a general generalized clause in $LP(X)$. A generalized literal $g$ of $\Phi$ is called a local extremely complex form, if

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

Published by Atlantis Press
Copyright: the authors 1123
(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.** Let \( \Phi_1, \Phi_2, \ldots, \Phi_n \) be general generalized clauses in \( \text{LP}(X) \), \( H_i(i = 1, 2, \ldots, n) \) the set of generalized literals in \( \Phi_i \), respectively. \( \alpha \in L \). If there exists \( g_i \in H_i \), such that \( \wedge_{i=1}^{n} g_i \leq \alpha \), then

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

is called an \( \alpha \)-generalized resolvent of \( \Phi_1, \Phi_2, \ldots, \Phi_n \), denoted by \( G = \text{R}_{(g-a)-g}(\Phi_1, \Phi_2, \ldots, \Phi_n) \).

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

1. \( D_i \in S (i = 1, 2, \ldots, m) \), or
2. There exist \( r_1, r_2, \ldots, r_k < i \), such that \( \text{R}_{(g-a)-g}(D_{r_1}, D_{r_2}, \ldots, D_{r_k}) = D_i \).

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

**Theorem 3.** Let \( S \) be a set of general generalized clauses in \( \text{LP}(X) \), \( \alpha \in L \), \( \{D_1, D_2, \ldots, D_m\} \) an \( \alpha \)-generalized resolution deduction from \( S \) to the general generalized clause \( D_m \). If \( D_m = \alpha \), then \( S \leq \alpha \).

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

**Definition 15.** Let \( g_1, g_2, \ldots, g_n \) be g-literals, a logical formula in \( \text{LF}(X) \) is called a general g-clause if these g-literals are connected by \( \land, \lor, \rightarrow, \leftarrow \) and \( \leftrightarrow \), denoted by \( \Phi(g_1, g_2, \ldots, g_n) \).

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

**Definition 16.** Let \( \Phi \) be a general g-clause in \( \text{LF}(X) \). If there exists a most general unifier \( \sigma \) of g-literals \( g_1, g_2, \ldots, g_m \) in \( \Phi \), then \( \Phi^\sigma \) is called a factor of \( \Phi \).

**Definition 17.** Let \( \Phi_1, \Phi_2, \ldots, \Phi_n \) be general g-clauses in \( \text{LF}(X) \). If \( \Phi_1 \) is a factor of \( \Phi_1 \) for g-literals \( g_{11}, g_{12}, \ldots, g_{1n_1} \), \( \Phi_2^r \) a factor of \( \Phi_2 \) for g-literals \( g_{21}, g_{22}, \ldots, g_{2n_2} \), and \( \Phi_n^r \) a factor of \( \Phi_n \) for g-literals \( g_{n1}, g_{n2}, \ldots, g_{nn_n}, \alpha \in L \). If \( \wedge_{i=1}^{n} g_i^r \leq \alpha \), then

\[ G = \vee_{i=1}^{n} \Phi_i^r(g_i^r = \alpha) \]

is called an \( \alpha \)-generalized resolvent of \( \Phi_1, \Phi_2, \ldots, \Phi_n \), denoted by \( G = \text{R}_{(g-a)-g}(\Phi_1, \Phi_2, \ldots, \Phi_n) \).

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

1. \( D_i \in S (i = 1, 2, \ldots, m) \), or
2. There exist \( r_1, r_2, \ldots, r_k < i \), such that \( \text{R}_{(g-a)-g}(D_{r_1}, D_{r_2}, \ldots, D_{r_k}) = D_i \).

**Theorem 5.** Let \( S \) be a set of general g-clauses in \( \text{LF}(X) \), \( \alpha \in L \), \( \{D_1, D_2, \ldots, D_m\} \) an \( \alpha \)-generalized resolution deduction from \( S \) to the general g-clause \( D_m \). If \( D_m = \alpha \), then \( S \leq \alpha \).

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

3. \( \alpha \)-Generalized lock resolution in lattice-valued logic

3.1. \( \alpha \)-Generalized lock resolution in \( \text{LP}(X) \)

**Definition 19.** Let \( \Phi \) be a general generalized clause in \( \text{LP}(X) \). \( \Phi \) is said to be locked if and only if for each generalized literal \( g \) in \( \Phi \), there exists a positive integer \( i \) such that \( i \) is the index of \( g \). This specific generalized generalized clause \( \Phi \) is called a locked general generalized clause.
Definition 20. Let $\Phi_1, \Phi_2, \ldots, \Phi_n$ be locked general
generalized clauses in $LP(X)$, $H_i$ the set of gener-
alized literals in $\Phi_i$, $\alpha \in L$. If there exists $g_i$ with
the minimal index in $H_i (i = 1, 2, \ldots, n)$, such that
$\bigwedge_{i=1}^{n} g_i \leq \alpha$, then
$$\Phi = \bigvee_{i=1}^{n} \Phi_i (g_i = \alpha)$$
is called an $\alpha$-generalized lock resolvent
of $\Phi_1, \Phi_2, \ldots, \Phi_n$, denoted by $\Phi =
R_{\alpha-g-L} (\Phi_1 (g_1), \Phi_2 (g_2), \ldots, \Phi_n (g_n))$.

$\alpha$-Generalized lock resolution is $\alpha$-lock resolu-
tion if the general generalized clause set is taken by
its conjunctive normal form.

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

1. $D_i \in S (i = 1, 2, \ldots, m)$, or
2. There exist $r_1, r_2, \ldots, r_k < i$, such that
$R_{\alpha-g-L} (D_1, D_2, \ldots, D_i) = D_i$.

Remark 3.

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

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

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

Proof. It follows directly by Theorem 3. \qed

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

Proof. We only convert $\Phi_1$ into its conjunctive nor-
mal form, i.e., $\Phi_1 = G_1 \land G_2 \land \cdots \land G_m$, where $G_i$
is a generalized clause, and $G_i = g_{i_1} \lor g_{i_2} \lor \cdots \lor g_{i_m}$, $g_{i_j}$
is a generalized literal ($1 \leq i \leq m; 1 \leq j \leq m$). With-
out loss of generality, suppose only the generalized
clause $G_1$ includes $g_{1_1}$, we denote $S = (g_{1_1} \lor G_0^1) \land
G_2 \land \cdots \land G_m \land \Phi_2 \land \cdots \land \Phi_n$, where $G_0^1$
is the dis-
junction of generalized literals in $G_1$ except for $g_{1_1}$.
Hence, $S = (g_{1_1} \lor G_0^1) \land G_2 \land \cdots \land G_m \land \Phi_2 \land \cdots \land \Phi_n$. Let
$S_0 = G_2 \land \cdots \land G_m \land \Phi_2 \land \cdots \land \Phi_n$, then $S =
(g_{1_1} \lor G_0^1) \land S_0 = (g_{1_1} \land S_0) \lor (G_0^1 \land S_0)$. Since $S \leq \alpha$, then
$G_0^1 \land S_0 \leq \alpha_1$. Therefore, $S_1 = \{\Phi (g_{1_1} = \alpha) | \Phi \in
S\} = (\alpha \land S_0) \lor (G_0^1 \land S_0) \leq \alpha \lor (G_0^1 \land S_0) \leq \alpha$. \qed

Theorem 9. (Weak completeness of ground $\alpha$-
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 $S \leq \alpha$, then there exists an $\alpha$-
generalized lock refutation of $S$.

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

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

Suppose that Theorem 9 is true for $K(S) < n$, then there exists at least a $\alpha$-generalized clause in $S$ which is not locked and unit. Since $S \leq \alpha$, then there exist generalized literals $g_1, g_2, \ldots, g_m (m \leq n)$ such that $g_1 \land g_2 \land \cdots \land g_m \leq \alpha$, where $g_i \in \Phi_i (1 \leq i \leq m)$. Let $g_1$ be a generalized literal with the largest index in $\Phi_1$, denote $S_1 = \{\Phi_1 (g_1 = \alpha) | \Phi_1 \in S\}$. By Proposition 8, we have $S_1 \leq \alpha$. Since the new substituted constant $\alpha$ has no index, then $K(S_1) < n$. By the induction hypothesis, there exists an $\alpha$-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 $\alpha$ in $\Phi_1$, then we get a new $\alpha$-generalized lock resolution deduction $D_1$ from $S$ to a general generalized clause $\Phi_1^0$. If all the generalized literals in $\Phi_1$ are not involved in $D_1^0$, then $\Phi_1^0$ is an $\alpha$-constant clause. Hence $D_1$ is also an $\alpha$-generalized lock refutation of $S$, i.e., Theorem 9 holds. Otherwise, $g_1$ has the largest index in $\Phi_1$, and the same generalized literals have the same indices, hence $g_1$ does not resolve on in $D_1$. Therefore, $\Phi_1^0$ only includes the generalized literal $g_1$, and $\Phi_1^0 (g_1 = \alpha)$ is an $\alpha$-constant clause.

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

Repeating the above steps from $S_2$ to $S_m$, we denote $S_m = \{\Phi_m (g_m = \alpha) | \Phi_m \in S\}$. Similarly, we have $S_m \leq \alpha$ and $K(S_m) < n$. By the induction hypothesis, there exists an $\alpha$-generalized lock refutation $D_m^0$ of $S_m$. We renew $D_m$ to $D_m$ by adding $g_m$ to $\Phi_m (g_m = \alpha)$, where $g_m$ is substituted by $\alpha$ in $\Phi_m$, then $D_m$ is an $\alpha$-generalized lock resolution deduction from $S$ to a general generalized clause $\Phi_m^0$. If $\Phi_m^0$ is an $\alpha$-constant clause, then $D_m$ is also an $\alpha$-generalized lock refutation of $S$, i.e., Theorem 9 holds. Otherwise, $\Phi_2^0$ is a general generalized clause, which only includes $g_m$, then $\Phi_m^0 (g_m = \alpha)$ is an $\alpha$-constant clause. Since $\Phi_i^0 (1 \leq i \leq m)$ only includes $g_i$, then $g_i$ has the minimal index of $\Phi_i^0$. Furthermore, from $g_1 \land g_2 \land \cdots \land g_m \leq \alpha$, it follows that $\bigwedge_{i=1}^m \Phi_i^0 (g_i = \alpha) = \alpha$ is an $\alpha$-generalized lock resolution of $\Phi_1^0, \Phi_2^0, \cdots, \Phi_m^0$. Therefore, we connect the $\alpha$-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 $\alpha$-generalized lock refutation of $S$. □

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

\begin{align*}
(1) & \quad x(x \rightarrow a_7) \land z(x \rightarrow y) \\
(2) & \quad 3x \land (4x \rightarrow y) \lor 5((x \rightarrow y) \rightarrow a_2)) \\
(3) & \quad 6(z \rightarrow a_7) \lor 7(x \rightarrow a_7)) \\
(4) & \quad 8(w \rightarrow z)' \\
(5) & \quad 2(x \rightarrow y) \\
(6) & \quad (a_6 \land 2(x \rightarrow y)) \lor 9((x \rightarrow a_7) \\
(7) & \quad 10(a_6 \land (4x \rightarrow y) \lor 5((x \rightarrow y) \rightarrow a_2)) \\
(8) & \quad 11(a_6 \land (4x \rightarrow y) \lor 5((x \rightarrow y) \rightarrow a_2)) \\
(9) & \quad 12(a_6 \land 2(x \rightarrow y) \lor 5((x \rightarrow y) \rightarrow a_2)) \\
(10) & \quad 13(a_6 \land 5((x \rightarrow y) \rightarrow a_2)) \\
(11) & \quad 14(a_6 \land 5((x \rightarrow y) \rightarrow a_2)) \\
(12) & \quad a_6
\end{align*}
However, 34 generalized clauses are generated for \( \alpha \)-generalized resolution. In this case, \( \alpha \)-generalized lock resolution improves the efficiency of \( \alpha \)-generalized resolution.

### 3.2. \( \alpha \)-Generalized lock resolution in \( LF(X) \)

**Definition 22.** Let \( \Phi \) be a general \( g \)-clause in \( LF(X) \). \( \Phi \) is said to be locked if and only if for each \( g \)-literal \( g \) in \( \Phi \), there exists a positive integer \( i \) such that \( i \) is the index of \( g \). This specific general \( g \)-clause \( \Phi \) 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 \( \Phi \) be a locked general \( g \)-clause in \( LF(X) \). If there exists a most general unifier \( \sigma \) of \( g \)-literals \( g_1, g_2, \cdots, g_m \) in \( \Phi \), then \( \Phi^\sigma \) is called a locked factor of \( \Phi \).

**Definition 24.** Let \( \Phi_1, \Phi_2, \cdots, \Phi_n \) be locked general \( g \)-clauses in \( LF(X) \), \( \Phi_i^\tau_i \) a factor of \( \Phi_i \) for \( g \)-literals \( g_{1i}, g_{2i}, \cdots, g_{ri} \), \( \Phi_i^\tau_i \) a factor of \( \Phi_i \) for \( g \)-literals \( g_{ni}, g_{ni+1}, \cdots, g_{ni+mi} \), \( \alpha \in L \). If there exists \( g_{i1} \) with the minimal index in \( \Phi_i^\tau_i(i = 1, 2, \cdots, n) \), such that \( \wedge_{i=1}^n g_{i1} \leq \alpha \), then

\[
\Phi = \bigvee_{i=1}^n \Phi_i^\tau_i(g_{i1} = \alpha)
\]

is called an \( \alpha \)-generalized lock resolvent of \( \Phi_1, \Phi_2, \cdots, \Phi_n \), 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 \( \alpha \)-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 \( \alpha \) lock resolution \( \delta \) 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 \( \tau \) be a substitution, the index of \( g \)-literal \( g^\tau \) in \( \Phi_i^\tau \) be the index of \( g \) in \( \Phi_i(i = 1, 2, \cdots, n) \). If \( P^0 \) is an \( \alpha \)-generalized lock resolvent of \( \Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0 \), then there exists an \( \alpha \)-generalized lock resolvent \( P \) of \( \Phi_1, \Phi_2, \cdots, \Phi_n \) and a substitution \( \lambda \) 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 \( \epsilon \), such that \( \Phi_1^0 = \Phi_1^\epsilon, \Phi_2^0 = \Phi_2^\epsilon, \cdots, \Phi_n^0 = \Phi_n^\epsilon \), and all the generalized literals in \( \Phi_i^\epsilon \) have the same indices with those in \( \Phi_i(i = 1, 2, \cdots, n) \). Let \( \Phi_i^\epsilon, \Phi_i^\epsilon, \cdots, \Phi_i^\epsilon \) be locked factors of \( \Phi_1^\epsilon, \Phi_2^\epsilon, \cdots, \Phi_n^\epsilon \) respectively, where \( \sigma_1 \) is a most general unifier \( g_{\sigma_1}^\epsilon, g_{\sigma_2}^\epsilon, \cdots, g_{\sigma_r}^\epsilon \) in \( \Phi_1^\epsilon \), \( \sigma_2 \) is a most general unifier \( g_{\sigma_2}^\epsilon, g_{\sigma_2}^\epsilon, \cdots, g_{\sigma_2}^\epsilon \) for \( \Phi_2^\epsilon \), \( \cdots, \sigma_n \) is a most general unifier \( g_{\sigma_n}^\epsilon, g_{\sigma_n}^\epsilon, \cdots, g_{\sigma_n}^\epsilon \) for \( \Phi_n^\epsilon \) in \( \Phi_i^\epsilon \) with \( g_{\sigma_i}^\epsilon \) having the minimal index in \( \Phi_i^\epsilon(i = 1, 2, \cdots, n) \). Hence, \( g_{\sigma_1}^\epsilon \) is a \( \alpha \)-generalized lock resolvent of \( \Phi_1^\epsilon, \Phi_2^\epsilon, \cdots, \Phi_n^\epsilon \) in \( \Phi_i^\epsilon \) with \( \wedge_{i=1}^n g_{\sigma_i}^\epsilon \leq \alpha \), then \( P^0 = R_{\alpha-g-L}(\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0) = \bigvee_{i=1}^n \Phi_i^{\epsilon, \sigma_i}(g_{\sigma_i}^\epsilon = \alpha) \).

Since \( \epsilon \cdot \sigma_1 \) is a unifier of \( g_{\sigma_1}^\epsilon, g_{\sigma_2}^\epsilon, \cdots, g_{\sigma_n}^\epsilon \), then there exists a most general unifier \( \lambda_i \) such that \( g_{\lambda_i}^\epsilon = g_{\lambda_i}^\epsilon = \cdots = g_{\lambda_i}^\epsilon \). Hence, there exists a substitution \( \beta \) such that \( \epsilon \cdot \sigma_1 = \lambda_i \cdot \beta \), where \( i = 1, 2, \cdots, n \). From the hypothesis that \( \Phi_1, \Phi_2, \cdots, \Phi_n \) have no common variables, it follows that \( \lambda_1, \lambda_2, \lambda_3, \lambda_n \) have no common variables, and \( \beta_1, \beta_2, \beta_3, \beta_n \) 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 \( i \in {1, 2, \cdots, n} \), \( g_{\lambda_i}^\epsilon = g_{\lambda_i}^\epsilon = \cdots = g_{\lambda_i}^\epsilon \). In this case, \( \wedge_{i=1}^n g_{\lambda_i}^\epsilon \leq \alpha \). Furthermore, all the generalized liter-
als in $\Phi^\alpha$ have the same indices with those in $\Phi(i = 1, 2, \cdots, n)$, hence $g_{i1}$ has the minimal index in $\Phi(i = 1, 2, \cdots, n)$. Therefore, $P = R_{a-g-L}(\Phi_1, \Phi_2, \cdots, \Phi_n) = \bigvee_{i=1}^n \Phi_i^0(g_{i1}^1 = \alpha)$.

Moreover,

$$P^0 = R_{a-g-L}(\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0) = \bigvee_{i=1}^n \Phi_i^{0\beta}(g_{i1}^1 = \alpha) = \bigvee_{i=1}^n \Phi_i^{\beta}(g_{i1}^1 = \alpha) = (P)^\beta$$

\[ \square \]

**Theorem 11.** (Soundness of $a$-generalized lock resolution) Let $S$ be a set of locked general $g$-clauses in $LF(X)$, where the same $g$-literals have the same indices. If $S \subseteq \alpha$, then there exists an $a$-generalized lock refutation of $S$.

**Proof.** It follows directly by Theorem 5. \[ \square \]

**Theorem 12.** (Weak completeness of $a$-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 $S \subseteq \alpha$, then there exists an $a$-generalized lock refutation of $S$.

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

4. Equivalent transformation of $a$-generalized lock resolution

**Theorem 13.** Suppose $S$ is a set of locked general $g$-clauses in $LF(X)$, $\alpha \in L$. There exists an $a$-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 $a$-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 $D_i(1 \leq i \leq m)$, 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 $a$-generalized lock resolvent $D_i(1 \leq i \leq m)$ in $LF(X)$, there exist locked general $g$-clauses $\Phi_1, \Phi_2, \cdots, \Phi_n$ such that $D_i = R_{a-g-L}(\Phi_1, \Phi_2, \cdots, \Phi_n)$, where $g_{1,2,\cdots,n}$ are the $a$-generalized lock resolved $g$-literals in $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively. Hence, $g_1 \land g_2 \land \cdots \land g_n \subseteq \alpha$, and $g_i$ has the minimal index in $\Phi_i(1 \leq i \leq m)$. Since $g_1 \land g_2 \land \cdots \land g_n \subseteq \alpha$, then for any interpretation $I_D = < D, \mu_D, \nu_D >$ in $LF(X)$, such that $\nu_D(g_1 \land g_2 \land \cdots \land g_n) \subseteq \alpha$. Let $D_1^0, D_2^0, \cdots, D_n^0$ be the ground instances of $g_1, g_2, \cdots, g_n$, respectively, for any valuation function $\nu_D$ of $\Phi_1, \Phi_2, \cdots, \Phi_n$ in $LP(X)$. Therefore, $g_{1,2,\cdots,n}$ has the minimal index of $\Phi_i^0(1 \leq i \leq n)$. Hence, $D_i^0 = R_{a-g-L}(\Phi_1^0, \Phi_2^0, \cdots, \Phi_n^0)$ is an $a$-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 $a$-generalized lock resolution). \[ \square \]

**Theorem 14.** Suppose $S$ is a set of locked general generalized clauses in $L_{V(nx2)}(X)$, $\alpha = (h_1, t)$. There exists a $(h_1, t)$-generalized lock resolution deduction $w = [D_1, D_2, \cdots, D_m]$ from $S$ to $D_m$ in $L_{V(nx2)}(X)$ if and only if there exists an $(a_1, a_2)$-generalized lock resolution deduction $w^0 = [D_1^0, D_2^0, \cdots, D_m^0]$ from $S^0$ to $D_m^0$ in $L_{V(nx2)}(X)$, where each of the locked generalized literals in $S^0$ and $D_i^0$ denote the restrictions of locked generalized literals in $S$ and $D_i(1 \leq i \leq m)$ on $L_{V(nx2)}(X)$, respectively.

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

Let $\phi$ be an isomorphic mapping as $\phi : L_{V(nx2)} \rightarrow L_{V(nx2)}$, then $\phi$ can be expanded to an isomorphic mapping from $L_{V(nx2)}(X)$ to $L_{V(nx2)}(X)$, denoted by $\phi_1$. Hence, for any formula $\Phi$ in $L_{V(nx2)}(X)$, $\Phi^* = \phi_1(\Phi)$, and $\Phi^*$ belongs to...
For each \( \alpha \)-generalized lock resolvent \( D_i(1 \leq i \leq m) \) in \( L_{n \times 2}P(X) \), there exist general generalized clauses \( \Phi_1, \Phi_2, \ldots, \Phi_n \) such that \( D_i = R_{\alpha - g_1}(\Phi_1, \Phi_2, \ldots, \Phi_n) \). Let \( g_1, g_2, \ldots, g_n \) be \((h_k,b_k)\)-generalized lock resolved literals in \( \Phi_1, \Phi_2, \ldots, \Phi_n \), respectively, then \( g_1 \land g_2 \land \cdots \land g_n \leq (a_k,b_k) \), and \( g_1, g_2, \ldots, g_n \) are generalized literals with the minimal indices in \( \Phi_1, \Phi_2, \ldots, \Phi_n \), respectively. Hence, for any valuation \( \gamma = \gamma_1 \land \gamma_2 \) in \( L_{n \times 2}P(X) \), we have \( \gamma(g_1 \land g_2 \land \cdots \land g_n) \leq (a_k,b_k) \). Then, we have \( \gamma(g_1(g_1 \land g_2 \land \cdots \land g_n)) \leq (a_k,b_k) \), and \( g_1(g_1 \land g_2 \land \cdots \land g_n) \leq a_k \), that is, \( \gamma_1(g_1) \) and \( \gamma_2(g_1) \). Denote \( g_1^* = \gamma_1(g_1) \), \( \Phi_i^* = \gamma_1(\Phi_i)(1 \leq i \leq m) \), i.e., \( g_1^* \) and \( \Phi_i^* \) are the restrictions of \( g_1 \) and \( \Phi_i \) on \( L_{n \times 2}(X) \), respectively. Then there exist generalized literals \( g_1^*, g_2^*, \ldots, g_n^* \) in \( \Phi_1^*, \Phi_2^*, \ldots, \Phi_n^* \), such that \( g_1^* \land g_2^* \land \cdots \land g_n^* \leq a_k \). Moreover, \( \Phi_i^* \) is syntactically equal to \( \Phi_i^* \) except for constants, hence all generalized literals of \( \Phi_i \) have the same indices with generalized literals of \( \Phi_i^* \), that is, \( g_1^* \) has the minimal index of \( \Phi_i^* \). Hence, \( D_i^* = R_{\alpha - g_1}(\Phi_1^*, \Phi_2^*, \ldots, \Phi_n^*) \) is an \( \alpha \)-generalized lock resolvent of \( \Phi_1^*, \Phi_2^*, \ldots, \Phi_n^* \), and \( D_i^* \) is the restriction of \( D_i \) on \( L_{n \times 2}(X) \).

Remark 4.

(1) Similarly equivalent transformations still hold for \( \alpha \)-generalized resolution deduction, i.e., \( \alpha \)-generalized resolution can be transformed from \( L_{n \times 2}(X) \) into \( L_{n \times 2}(X) \), which can also simplify the complexity of \( \alpha \)-generalized resolution.

(2) For \( \alpha \)-generalized lock resolution in \( L_{n \times 2}(X) \), we can equivalently transform it into that in \( L_{n \times 2}(X) \) by Theorem 13, and further to that in \( L_{n \times 2}(X) \) by Theorem 14, and finally to \( L_n(X) \) by Theorem 16 which takes truth-values in a Łukasiewicz implication algebra on a finite chain \( L_n \), that is, for discussing \( \alpha \)-generalized lock resolution in \( L_{n \times 2}(X) \), we only need to discuss that in \( L_n(X) \).
5. Compatibilities of α-generalized lock resolution

5.1. α-Generalized linear semi-lock resolution method in $L_n\mathbb{P}(\mathbb{X})$

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 $LF(X), \alpha \in L, \Phi_0$ is a general g-clause in $S$. An $\alpha$-generalized resolution deduction $w = \{D_1, D_2, \cdots, D_m\}$ is called an $\alpha$-generalized linear resolution deduction from $S$ with the top general g-clause $\Phi_0$ to the general g-clause $D_m$ if it satisfies

1. $D_{i+1}$ is the $\alpha$-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 $LF(X), \alpha \in L, \Phi_0$ is a locked general g-clause in $S$. An $\alpha$-generalized linear resolution deduction $w = \{D_1, D_2, \cdots, D_m\}$ is called an $\alpha$-generalized linear lock resolution deduction from $S$ with the top general g-clause $\Phi_0$ 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 $\alpha$-generalized lock resolvent of $D_i$ (center g-clause) and $B_i$ (side g-clause). $w$ is called an $\alpha$-generalized linear semi-lock resolution deduction from $S$ with the top g-clause $\Phi_0$ to the g-clause $D_m$ if the $\alpha$-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 $L_n\mathbb{P}(\mathbb{X})$, where the same generalized literals have the same indices in $\Phi_i(1 \leq i \leq n)$, all the indices of generalized literals in $\Phi_i$ are less than those in $\Phi_i(j < i)$. If $S \subseteq \alpha$, then $S - \{\Phi_1\}$ is $\alpha$-satisfiable, there exists an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $\Phi_1$.

Proof. If $S$ only includes a general generalized clause $\Phi_0$, then $\Phi_0 \subseteq \alpha$ by $S \subseteq \alpha$. Hence, $\Phi_0$ has an $\alpha$-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., $S = g_1 \land g_2$. Obviously, there exists an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $g_1$.

Suppose Theorem 17 is true for $m(S) < n$. Since $S \subseteq \alpha$, then there exist generalized literals $g_1, g_2, \cdots, g_m$ in $\Phi_1, \Phi_2, \cdots, \Phi_m$, respectively, such that $g_1 \land g_2 \land \cdots \land g_m \subseteq \alpha(m \leq n)$. Without loss of generality, let $g_i$ have the largest index of $\Phi_i(1 \leq i \leq m)$, and $S$ the minimum $\alpha$-unsatisfiable set which includes $\Phi_i$. Denote $S_1 = \{\Phi_i(g_1 = \alpha) | \Phi_i \in S\}, \Phi_0 = \Phi_1(g_1 = \alpha)$. By Proposition 8, we have $S_1 \subseteq \alpha$. Obviously, $m(S_1) < n$. From $S_1 - \{\Phi_0\} = S - \{\Phi_1\}$, it follows that $S_1 - \{\Phi_0\}$ is $\alpha$-satisfiable. By the induction hypothesis, there exists an $\alpha$-generalized linear semi-lock refutation $D_1^0$ of $S_1$ with the top clause $\Phi_0$.

We renew $D_1^0$ to $D_1$ by adding $g_1$ to $\Phi_0$, where $g_1$ is substituted by $\alpha$, then $D_1$ is an $\alpha$-generalized linear semi-lock resolution deduction from $S$ with the top clause $\Phi_0$ to the general clause $\Psi_1$. For $\Psi_1$, two cases follow: $\Psi_1$ is either an $\alpha$-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 $S_2 = \{\Phi_i(g_1 = \alpha) | \Phi_i \in S\}, \Phi_2 = \Phi_1(g_1 = \alpha)$, then $S_2 \subseteq \alpha$ and $S_2 - \{\Phi_2\}$ is $\alpha$-satisfiable $(2 \leq i \leq m)$. Furthermore, $m(S_2) < n$. By the induction hypothesis, there exists an $\alpha$-generalized linear semi-lock refutation $D_2^0$ of $S_2$ with the top clause $\Phi_0$. We only renew $D_2^0$ to $D_2$ by adding $g_i$ to $\Phi_0$, where $g_i$ is substituted by $\alpha$ in side clauses of $D_2^0$, then $D_2$ is an $\alpha$-generalized linear semi-lock resolution deduction from $S$ with the top clause $\Phi_0$ to $\Psi_i$. For $\Psi_i$, two cases follow: $\Psi_i$ is either an $\alpha$-constant clause or a general generalized clause which only includes $g_i$, and $\Psi_i(g_i = \alpha) = \alpha$.

For the resolvent $\Psi_i$ in $D_1$, if $\Psi_i$ is an $\alpha$-constant clause, then $D_1$ is also an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $\Phi_1$, i.e., Theorem 17 holds. Otherwise, $\Psi_1$ is a general gener-
alized clause, which only includes $g_1$, and $\Psi_1(g_1 = \alpha) = \alpha$. In this case, resolve on $\Psi_1$, $\Phi_2$, $\ldots$, $\Phi_m$, and get $\Psi_1' = R(g_1 - \alpha)(\Psi_1(g_1), \Phi_2(g_2), \ldots, \Phi_m(g_m)) = \Psi_1(g_1 = \alpha) \lor \Phi_2(g_2 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$, hence we get an $\alpha$-generalized linear semi-lock deduction $D_1^g$ of $S$ with the top clause $\Phi_1$ to $\Phi_2(g_2 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$. Hence, we add $\Phi_3(g_3 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ to the top clause $\Phi_2(g_2 = \alpha)$ in $D_2$. Since the indices of all generalized literals in $\Phi_i$ are less than those of $\Phi_j(i < j)$, then $\Phi_3(g_3 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ can not be resolved in $D_2$. Hence, we get an $\alpha$-generalized linear semi-lock resolution deduction $D_2^g$ of $S$ with the top clause $\Phi_2(g_2 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ to the generalized clause $\Psi_2^g = \Psi_2 \lor \Phi_3(g_3 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$. Similarly, if $\Psi_2$ is an $\alpha$-constant clause, then $\Psi_2^g = \Phi_3(g_3 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$. Otherwise, $\Psi_2$ is a generalized clause which only includes $g_2$ and $\Psi_2(g_2 = \alpha) = \alpha$. In this case, resolve on $\Phi_1, \Psi_2, \Phi_3, \ldots, \Phi_m$, and get $\Psi_2' = R(g_2 - \alpha)(\Psi_1(g_1), \Psi_2(g_2), \ldots, \Phi_m(g_m)) = \Psi_1(g_1 = \alpha) \lor \Psi_2(g_2 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$, it is also an $\alpha$-generalized linear semi-lock deduction $D_2^g$ of $S$ with the top clause $\Phi_2(g_2 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ to $\Psi_2^g$. Hence, we add $\Phi_4(g_4 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ to $\Phi_3(g_3 = \alpha)$ in $D_3$. Since all the indices of generalized literals in $\Phi_i$ are less than those of $\Phi_j(i < j)$, then $\Phi_4(g_4 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$ can not be resolved in $D_3$. Hence, we get an $\alpha$-generalized linear semi-lock resolution deduction $D_3^g$ of $S$ with the top clause $\Psi_2^g$ to the general generalized clause $\Psi_3^g = \Psi_3 \lor \Phi_4(g_4 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$. Similarly, if $\Psi_3$ is an $\alpha$-constant clause, then $\Psi_3^g = \Phi_4(g_4 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$. Otherwise, $\Psi_3$ is a generalized clause, which only includes $g_3$, and $\Psi_3(g_3 = \alpha) = \alpha$. Hence, we resolve on $\Psi_1, \Psi_2, \Phi_3, \Phi_4, \ldots, \Phi_m$, and get $\Psi_3' = R(g_3 - \alpha)(\Psi_1, \Psi_2, \Phi_3, \Phi_4, \ldots, \Phi_m) = \Psi_1(g_1 = \alpha) \lor \Psi_2(g_2 = \alpha) \lor \Psi_3(g_3 = \alpha) \lor \Phi_4(g_4 = \alpha) \lor \cdots \lor \Phi_m(g_m = \alpha)$, it is also an $\alpha$-generalized linear semi-lock deduction $D_3^g$ of $S$ with the top clause $\Psi_2^g$ to $\Psi_3^g$.

Repeat the similar steps above, we get $\Psi_{m-1}^g = R(g_m - \alpha)(\Psi_1, \Psi_2, \ldots, \Phi_{m-1}, \Phi_m) = \Psi_1(g_1 = \alpha) \lor \Psi_2(g_2 = \alpha) \lor \cdots \lor \Psi_{m-1}(g_{m-1} = \alpha) \lor \Phi_m(g_m = \alpha)$, it is also an $\alpha$-generalized linear semi-lock deduction $D_{m-1}^g$ of $S$ with the top clause $\Psi_{m-2}$ to $\Psi_{m-1}$. Since $\Psi_{m-1}^g = \Phi_m$ and $D_m$ is an $\alpha$-generalized linear semi-lock resolution deduction from $S$ with the top clause $\Phi_m^0$ to $\Phi_m$. If $\Psi_m^g$ is an $\alpha$-constant clause, then connecting the deductions $D_1^g, D_2^g, \ldots, D_{m-1}^g, D_m$, we get an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $\Phi_1$, i.e., Theorem 17 holds. Otherwise, $\Psi_m^g$ is a generalized clause which only includes $g_m$, and $\Psi_m(g_m = \alpha) = \alpha$. In this case, we resolve on $\Psi_1, \Psi_2, \Psi_3, \ldots, \Phi_{m-1}, \Phi_m$, and get $\Psi_m' = R(g_m - \alpha)(\Psi_1, \Psi_2, \Psi_3, \ldots, \Phi_{m-1}, \Phi_m) = \Psi_1(g_1 = \alpha) \lor \Psi_2(g_2 = \alpha) \lor \Psi_3(g_3 = \alpha) \lor \cdots \lor \Phi_{m-1}(g_{m-1} = \alpha) \lor \Phi_m(g_m = \alpha) = \alpha$, hence we get an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $\Phi_1$. Therefore, we connect the $\alpha$-generalized linear semi-lock resolution branches $D_1^g, D_2^g, \ldots, D_m^g$, and denote $D^g = D_1^g \cup D_2^g \cup \cdots \cup D_m^g$, then $D^g$ is an $\alpha$-generalized linear semi-lock refutation of $S$ with the top clause $\Phi_1$.

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

**Proof.** Similar to the proof of Theorem 12, the completeness of $\alpha$-generalized linear semi-lock resolution in $L_n(F(X))$ follows.

5.2. An algorithm for $\alpha$-generalized linear semi-lock resolution in $L_n(P(X))$

In $\alpha$-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 $g_1, g_2, \ldots, g_m$ be generalized literals in $L(P(X),g_1,g_2,\ldots,g_m)$, $g_1 \land g_2 \land \cdots \land g_k \notin \alpha$.

**Example 2.** Let $L_9 = \{a_i\vert i \leq 9\}$ be a Łukasiewicz implication algebra, $x, y$ propositional variables in
Step 0. Given generalized literals \( g_1, g_2, \cdots, g_m \) in \( L(P(X)) \), \( \alpha = a_k \).

Step 1. Judge \( g_1 \land g_2 \land \cdots \land g_m \leq a_k \). If it does not satisfy, then stop, \( \text{Res}_m(g_1, g_2, \cdots, g_m, a_k) = 0 \).

Step 2. \( k = 1 \).

Step 3. For any \( k \) generalized literals \( \{g_{i_1}, g_{i_2}, \cdots, g_{i_k}\} \subset \{g_1, g_2, \cdots, g_m\} \), judge \( g_{i_1} \land g_{i_2} \land \cdots \land g_{i_k} \leq a_k \). If it satisfies, then stop, \( \text{Res}_m(g_{i_1}, g_{i_2}, \cdots, g_{i_m}, a_k) = 0 \).

Step 4. If \( k < m - 1 \), then \( k = k + 1 \), go to Step 3. Otherwise, stop, \( \text{Res}_m(g_1, g_2, \cdots, g_m, a_k) = 1 \).

Algorithm 1 can be seen as a function, i.e., \( \text{Res}_m(g_1, g_2, \cdots, g_m, a_k) \). Let \( g_1, g_2, \cdots, g_m \) be generalizable literals in \( L(P(X)) \). \( g_1, g_2, \cdots, g_m \) are \( a_k \)-minimum resolved if \( \text{Res}_m(g_1, g_2, \cdots, g_m, a_k) = 1 \). Otherwise, it returns 0.

According to Algorithm 1, an algorithm for \( \alpha \)-generalized linear semi-lock resolution follows.

Algorithm 2

Step 0. (Initiation) Let \( S \) be a set of locked general \( \alpha \)-generalized clauses \( \Phi_1, \Phi_2, \cdots, \Phi_n \). Assign to each occurrence of \( \alpha \)-generalized literal a positive integer in \( \Phi_i \), the same \( \alpha \)-generalized literals have the same indices, and all the indices of \( \alpha \)-generalized literals in \( \Phi_i \) 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 \( \alpha \)-generalized literal with the minimal index in \( \Phi \). \( H \) the set of \( \alpha \)-generalized literals in \( S - \{\Phi\} \).

Step 2. Let \( n_0 \) be the number of \( \alpha \)-generalized literals in \( H \). For \( i_0 = 1 \) to \( n_0 \), if there exist \( i_0 \) \( \alpha \)-generalized literals \( g_{j_1}, g_{j_2}, \cdots, g_{j_{i_0}} \) \( (g_{j_i} \in \Phi_j) \), such that \( \text{Res}_m(g_{j_1}, g_{j_2}, \cdots, g_{j_{i_0}}, a_k) = 1 \), then \( \Phi_m = R_{a_k - L}(\Phi, \Phi_{i_1}, \Phi_{i_2}, \cdots, \Phi_{i_{i_0}}) \). If \( \Phi_m = a_k \), then \( S \) \( a_k \)-satisfiable.

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

Step 4. If \( c \leq c_0 \), then go to Step 1. Otherwise, stop, \( S \) \( a_k \)-satisfiable.

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

Theorem 19. (Soundness) If Algorithm 2 terminates, then the \( a_k \)-unsatisfiability of \( S \) can be judged in \( L(P(X)) \).

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 \( \Phi_m \), such that \( \Phi_m = a_k \), then by the soundness of \( a_k \)-generalized linear semi-lock resolution in \( L(P(X)) \), we have \( S \leq a_k \). Otherwise, if no \( a_k \)-generalized lock resolvents exists, then \( S \) \( a_k \)-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 \( S \leq a_k \) in \( L(P(X)) \), then Algorithm 2 terminates in Step 2.

Proof. If \( S \leq a_k \), then by Algorithm 2 and the completeness of \( a_k \)-generalized linear semi-lock resolution in \( L(P(X)) \), 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 (\( \mathcal{L}(\alpha) \times \mathcal{L}(\alpha) \), i.e., \( \alpha \)-generalized lock resolution and \( \alpha \)-generalized linear semi-lock resolution including their concepts, soundness and completeness. We equivalently transformed \( \alpha \)-generalized lock resolution in \( \mathcal{L}(\alpha) \times \mathcal{L}(\alpha) \) to that in \( L(P(X)) \), it greatly simplifies its complexity for \( \mathcal{L}(\alpha) \times \mathcal{L}(\alpha) \). In order to further
improve the efficiency of $\alpha$-generalized lock resolution, we discussed $\alpha$-generalized linear semi-lock resolution by combining it with $\alpha$-generalized linear resolution. The further research will be concentrated on discussing how to assign appropriate indices to the generalized literals in $\alpha$-generalized lock resolution, and exploring some theoretical or practical applications for $\alpha$-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