International Journal of Computational Intelligence Systems, Vol. 8, Supplement 1 (2015), 75-84

α-Resolution Method for Lattice-valued Horn Generalized Clauses in Lattice-valued Propositional Logic Systems

Weitao Xu [1] *, Wenqiang Zhang [2], Dexian Zhang [3], Yang Xu [4], Xiaodong Pan [5]

[1,2,3] College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China
E-mail: hnxmxwt@163.com, wqzhang@haut.edu.cn, zdxzzit@hotmail.com
[4,5] School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China
E-mail: xuyang@home.swjtu.edu.cn, xdpan1@163.com

Received 25 February 2015

Accepted 27 October 2015

Abstract

In this paper, an α-resolution method for a set of lattice-valued Horn generalized clauses is established in lattice-valued propositional logic system $\mathscr{L}P(X)$ based on lattice implication algebra. Firstly, the notions of lattice-valued Horn generalized clause, normal lattice-valued Horn generalized clause and unit lattice-valued Horn generalized clause are given in $\mathscr{L}P(X)$. Then, the α-resolution of two lattice-valued Horn generalized clauses is represented in $\mathscr{L}P(X)$. It indicates the reasoning rules in a resolution process, which aims at deleting α-resolution literals and obtaining a resolvent. Finally, we build an α-resolution algorithm for a set of lattice-valued Horn generalized clauses in $\mathscr{L}P(X)$. It provides a foundation for automated reasoning in lattice-valued first-order logic system and an application for designing an inference system in the field of intelligent decision support.

Keywords: automated reasoning; lattice-valued logic; α-resolution; lattice-valued Horn g-clause; lattice implication algebra.

1. Introduction

In the real world, human intelligence actions are always involved with uncertain information processing, hence it is of important significance in AI that how to make the computer simulate human being to deal with uncertainty information. As one of the research fields in AI, automated reasoning plays an important role for achieving the intelligent computing reasoning in intelligent or complex systems.

One approach of automated theorem proving is resolution and its variants. Since the resolution principle is presented by Robinson in 1965[1], resolution-based automated reasoning has been extensively studied in the context of finding natural and efficient proof systems to support a wide spectrum of computational tasks.

In classical logic, based on Robinson's resolution principle, many resolution methods are studied, and a number of important applications of such systems have been found in some areas such as AI, logic programming, problem solving and question answering systems, and so on [2,3]. Specially, there are three typical resolution methods such as semantic resolution, linear resolution and lock resolution. Liu deeply studied these resolution methods and ex-

*Corresponding Author.

tended them to the set of the generalized clauses [4]. Lu also presented the resolution principle for a specific set, which is the set of Horn clauses in classical logic [5]. However, these classical resolution principles or methods based on classical logic are easy to deal with certain problem. In fact, there exists much uncertain information or knowledge in the real world. Because the real world is dealing with uncertainty, it is difficult to design any intelligent system based on traditional logic. Hence, the area of automated reasoning based on non-classical logic (especially multi-valued logic and fuzzy logic) has drawn many researchers' attention.

From the viewpoint of symbolism, it is highly necessary to study and establish a logic foundation for automated reasoning. Lattice-valued logic, which is an important kind of non-classical logic, plays an important role in dealing with comparability and incomparability. In order to establish the theories and methods to simultaneously deal with fuzziness and incomparability of processed object itself and uncertainty in the course of information processing, Xu presented lattice implication algebra by combining lattice with implication algebra [6]. Subsequently, Xu et al. established lattice-valued propositional logic system $\mathscr{L}P(X)$ and lattice-valued first-order logic system $\mathscr{L}F(X)$ based on lattice implication algebra [7,8,10]. These logic systems, which have not only stick syntax proof but also sound semantic interpretation, provide a scientific and reasonable logical foundation for intelligent information processing and theorem automated proving.

In the frame of lattice-valued propositional logic system $\mathscr{L}P(X)$, Xu et al. established α-resolution principle for the generalized clauses[9,10,17]. The α-resolution principle provides a crucial foundation to construct resolution method for automated reasoning. Moreover, Xu et al. further studied the properties of generalized literals, and the α-resolution determination table under 49 cases of any quasi-regular generalized literals and constants are given along with the proofs of all the cases in the a-resolution determination table [11]. These can provide an important resolution foundation in the resolution process. Liu et al. established a new resolution strategy based on lattice-valued logic, and constructed an automated reasoning algorithm [12]. Liu et al. aimed at the resolution principle for the Pavelka type fuzzy logic, and used this resolution-like principle to Horn clauses with truth-values in an enriched residuated lattice and consider the L-type fuzzy Prolog [13]. Liu et al. also proposed Lukasiewicz implication resolution and applied to the sets of Horn clauses in residuated lattice [14]. Tang et al. introduces an automatic Web service composition method based on logical inference of Horn clauses and Petri nets, and the Web service composition problem is transformed into the logical inference problem of Horn clauses by exploring the dependency relations among services [16]. So far, there have been a lot of excellent results and many research areas have been developed on automated reasoning, but most of them don't involve lattice-valued logic and gradational resolution level.

This paper is organized as follows: Section 2 as a preliminary gives an overview of some basic concepts of lattice implication algebra, basic concepts of α-resolution principle in lattice-valued propositional logic, where some relevant works are reviewed. Section 3 as a major work proposes α-resolution method for lattice-valued Horn generalized clauses in lattice-valued propositional logic system $\mathscr{L}P(X)$. An algorithm based on the present method is constructed in Section 4. Concluding remarks and future researches are presented in Section 5.

2. Preliminaries

In this section we review some essential conceptions about lattice implication algebra, α-resolution principle based on lattice-valued propositional logic system $\mathscr{L}P(X)$.

Definition 1. [10] (Lattice implication algebra) 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 $\rightarrow: L\times L \rightarrow L$ be a mapping. $\mathscr{L}=(L,\vee,\wedge,',\rightarrow,O,I)$ is called a lattice implication algebra if the following conditions hold for any $x,y,z\in L$,

(1) $x\rightarrow(y\rightarrow z)=y\rightarrow(x\rightarrow z)$;

(2) $x \to x = I$;

(3) $x \to y = y' \to x'$;

(4) $x \to y = y \to x = I$ implies $x = y$;

(5) $(x \to y) \to y = (y \to x) \to x$;

(6) $(x \vee y) \to z = (x \to z) \wedge (y \to z)$;

(7) $(x \wedge y) \to z = (x \to z) \vee (y \to z)$.

Example 1. [10] Let $L = \{O, a, b, c, d, I\}$ and $O' = I$, $a' = c$, $b' = d$, $c' = a$, $d' = b$, $I' = O$, the Hasse diagram of L be defined as Fig.1 and its implication operator be defined as Table 1, then $(L, \vee, \wedge, ', \to, O, I)$ is a lattice implication algebra, denoted by $\mathscr{L}_6$.

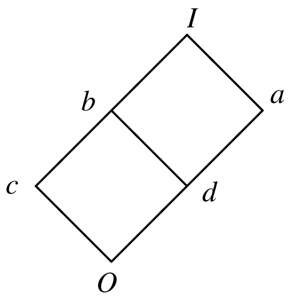

Fig. 1 Hasse diagram of $\mathscr{L}_6$

Table 1 Implication operator of $\mathscr{L}_6$

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

Definition 2. [18] Let $AD_n = \{a_1, a_2, \cdots, a_n\}$ be a set with n modifiers and $a_1 < a_2 < \cdots < a_n$, $MT = \{f, t\}$ be a set of meta truth values, $f < t$. Denote $L_{V(n\times 2)} = AD_n \times MT$. Define a mapping g as

$$g : L_{V(n\times 2)} \to \mathscr{L}_n \times \mathscr{L}_2,$$

and

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

then g is bijection, denote its inverse mapping as g^{-1}. For any $x, y \in L_{V(n\times 2)}$, define

$$x \vee y = g^{-1}(g(x) \vee g(y)),$$
$$x \wedge y = g^{-1}(g(x) \wedge g(y)),$$
$$x' = g^{-1}((g(x)),$$
$$x \to y = g^{-1}((g(x) \to g(y)).$$

We call $\mathscr{L}_{V(n\times 2)} = (L_{V(n\times 2)}, \vee, \wedge, ', \to, (a_n, f), (a_n, t))$ a linguistic truth-valued lattice implication algebra generated by AD_n and MT, its elements are called linguistic truth-values, and g is an isomorphic mapping from $(L_{V(n\times 2)}, \vee, \wedge, ', \to, (a_n, f), (a_n, t))$ to $\mathscr{L}_n \times \mathscr{L}_2$ (see Fig. 1).

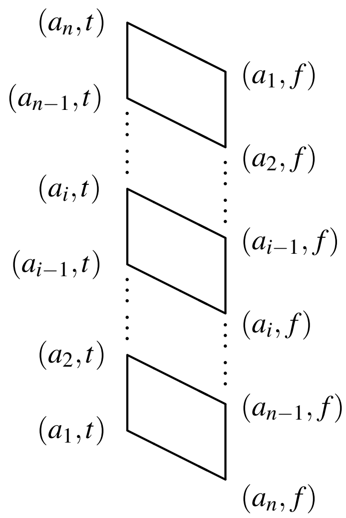

Fig. 1: Hasse diagram of linguistic truth-valued lattice implication algebra $\mathscr{L}_{V(n\times 2)}$

In the above definition, both $\mathscr{L}_n$ and $\mathscr{L}_2$ are Lukasiewicz implication algebras, i.e., $\mathscr{L}_n = (L_n, \vee_{(L_n)}, \wedge_{(L_n)}, '^{(L_n)}, \to_{(L_n)}, d_1, d_n)$, $\mathscr{L}_2 = (L_2, \vee_{(L_2)}, \wedge_{(L_2)}, '^{(L_2)}, \to_{(L_2)}, b_1, b_2)$. $\mathscr{L}_n \times \mathscr{L}_2$ is a lattice implication algebra generated by $\mathscr{L}_n$ and $\mathscr{L}_2$, i.e., $\mathscr{L}_n \times \mathscr{L}_2 = (L_n \times L_2, \vee, \wedge, ', \to, (d_1, b_1), (d_n, b_2))$.

Example 2. [18] Let AD_9={Slightly(Sl), Somewhat(So), Rather(Ra), Almost(Al), Exactly(Ex), Quit(Qu), Very(Ve), Highly(Hi), Absolutely(Ab)}, MT={False(F), True(T)}, and $Sl < So < Ra < Al < Ex < Qu < Ve < Hi < Ab$, $F < T$. According Definition 2, we can construct a linguistic truth-valued lattice implication algebra with 18 elements, denoted by $\mathscr{L}_{V(9\times 2)} = (L_{V(9\times 2)}, \vee, \wedge, ', \rightarrow, (Ab,F), (Ab,T))$.

Definition 3. [10] Let X be a set of propositional variables, L be a lattice implication algebra, and $T = L\bigcup\{', \rightarrow\}$ be a type with $ar(') = 1$, $ar(\rightarrow) = 2$, and $ar(a) = 0$ for any $a \in L$, where $ar : T \rightarrow N$ is a mapping, and N is a nonnegative integer set. The propositional algebra of the lattice-valued propositional calculus on the propositional variables is a free T algebra on X, denoted by $LP(X)$.

Definition 4. [10] A mapping $v : LP(X) \rightarrow L$ is called a valuation of $LP(X)$ if it is a T-homomorphism.

Definition 5. [10] Let $F, G \in LP(X)$. If $v(F) < v(G)$ for any valuation v of $LP(X)$, we say that F is always less than G, denoted by $F < G$. F and G are equivalent propositions and denoted by $F = G$, if $v(F) = v(G)$ for any valuation v of $LP(x)$.

Definition 6. [10] A lattice-valued propositional logic formula F is called an extremely simple form(ESF), if a lattice-valued propositional logic formula F^* obtained by deleting any constant, literal or implication term occurring in F is not equivalent to F.

Definition 7. [10] A lattice-valued propositional logic formula is called an indecomposable extremely simple form (IESF), if

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

(2) For any $G \in \mathscr{F}$, if $G \in \overline{F}$ in $\overline{LP(X)}$, then G is an ESF containing connectives $\rightarrow$ and $'$ at most, where $\mathscr{F}$ is a set of formulae in $LP(X)$.

Definition 8. [10] All the constants, literals and $IESF_S$ are called generalized literals.

Definition 9. [10] A lattice-valued propositional logical formula G is called a generalized clause (phrase) if G is a formula of the form

$$G = g_1 \vee \cdots \vee g_i \vee \cdots \vee g_n$$
$$\text{or } G = g_1 \wedge \cdots \wedge g_i \wedge \cdots \wedge g_n,$$

where g_i are generalized literals, $i = 1, \cdots, n$.

Definition 10. [10] Let $F \in LP(X)$, $\alpha \in L$. F is called $\alpha - false$, if for any valuation v of $LP(X)$, such that $v(F) \leqslant \alpha$.

Remark 1. If a generalized clause is α-false, then it is called an α-empty clause (for short, denoted by $\alpha - \diamond$).

Definition 11. [9] In lattice-valued propositional logic system $\mathscr{L}P(X)$, let $\mathscr{L}$ be a lattice implication algebra, $\alpha \in L$, G_1 and G_2 two generalized clauses of the form

$$G_1 = g_1 \vee \cdots \vee g_i \vee \cdots \vee g_m,$$
$$G_2 = h_1 \vee \cdots \vee h_j \vee \cdots \vee h_n,$$

where $g_i (i = 1, 2, \cdots, m)$ and $h_j (j = 1, 2, \cdots, n)$ are generalized literals in G_1 and G_2 respectively. If $g_i \wedge h_j \leqslant \alpha$, then

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

is called an α-resolvent of G_1 and G_2, denoted by $R_\alpha(G_1, G_2)$, and (g_i, h_j) is called an α-resolution pair, denoted by $(g_i, h_j) - \alpha$.

Theorem 1. *[9] Suppose a generalized conjunctive normal form $S = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ in $LP(X)$, $\alpha \in L$, $D_1, D_2, \cdots, D_m$ is an α-resolution deduction from S to a generalized clause D_m. If D_m is $\alpha - \diamond$, then $S \leqslant \alpha$, i.e., if $D_m \leqslant \alpha$, then $S \leqslant \alpha$.*

Theorem 2. *[9] Let S be a generalized conjunctive normal form in $LP(X)$, $\alpha \in L$, α a daul numerator and $\bigvee_{\alpha \in L}(a \wedge a') \leqslant \alpha < I$. Suppose that there exists $\beta \in L$ such that $\beta \wedge (\beta \rightarrow \beta') \nleqslant \alpha$. If $S \leqslant \alpha$, then there exists an α-resolution deduction from S to $\alpha - \diamond$.*

We refer the readers to [6,7,8,9,10] for more details of the concepts and properties about lattice implication algebra, lattice-valued propositional logic system $\mathscr{L}P(X)$ and α-resolution principle based on $\mathscr{L}P(X)$.

3. α-resolution method for a set of lattice-valued Horn generalized clauses

In the following, generalized literal and generalized clause are denoted by g-literal and g-clause respectively.

Definition 12. Let $\mathscr{L}P(X)$ be lattice-valued propositional logic system, p is called a positive literal, then p' is called a negative literal.

Definition 13. Let $\mathscr{L}P(X)$ be lattice-valued propositional logic system, g is called a positive g-literal, then g' is called a negative g-literal.

Example 3. Let p and q be positive literals in lattice-valued propositional logic system $\mathscr{L}P(X)$, then

(1) $p \to q$, $p' \to q$, $p' \to q'$, $p \to q'$ are also positive g-literals.

(2) $(p \to q)'$, $(p' \to q)'$, $(p' \to q')'$, $(p \to q')'$ are called negative g-literals.

Remark 2. In $\mathscr{L}P(X)$, a positive literal is still called a positive g-literal, and a negative literal is still called a negative g-literal.

Definition 14. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let r be a positive g-literal, and $h_1, h_2, \cdots, h_m$ are the negative g-literals, the clauses with at most one positive g-literal of the following form

$$h_1 \vee h_2 \vee \cdots \vee h_m \vee r, \quad or \quad h_1 \vee h_2 \vee \cdots \vee h_m, \quad or \quad r$$

are called lattice-valued Horn generalized clauses, shortly for lattice-valued Horn g-clause.

Example 4. In lattice-valued propositional logic system $\mathscr{L}_6P(X)$, let p, q, r be the literals, $a, b, c \in L_6$, and

$$H_1 = (p' \to b)' \vee r,$$

$$H_2 = (p \to a)' \vee (r \to p) \vee (r \to (p \to q))',$$

$$H_3 = c \to r,$$

then H_1, H_2, H_3 are lattice-valued Horn g-clauses.

Definition 15. In lattice-valued propositional logic system $\mathscr{L}P(X)$, if a lattice-valued Horn g-clause contains only one positive g-literal, then it is called unit lattice-valued Horn g-clause.

Example 5. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let p, q, r be the literals, $a \in L$, and

$$H_4 = p \to q,$$

$$H_5 = p \to q',$$

$$H_6 = p \to a,$$

$$H_7 = (p \to q) \to r,$$

$$H_8 = r,$$

then H_4, H_5, H_6, H_7, H_8 are unit lattice-valued Horn g-clauses.

Definition 16. In a lattice-valued Horn g-clause, if the rightmost g-literal is a positive g-literal, then it is called a normal lattice-valued Horn g-clause.

Example 6. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let p, q, r be the literals, and

$$H_9 = (p \to q)' \vee (r \to (p \to q))' \vee (r \to p),$$

$$H_{10} = (p' \to q)' \vee (q \to r),$$

then H_9, H_{10} are normal lattice-valued Horn g-clauses.

Remark 3. Obviously, a unit lattice-valued Horn g-clause is also a normal lattice-valued Horn g-clause.

Definition 17. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let S be a set of the g-clauses. S is called a set of lattice-valued Horn g-clauses if every g-clause in S is lattice-valued Horn g-clause.

Definition 18. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let H_1 and H_2 be lattice-valued Horn g-clauses. The resolvent of H_1 and H_2 are defined as follows.

Case 1: $H_1 = h_1 \vee h_2 \vee \cdots \vee h_m \vee r_1$, $H_2 = g_1 \vee g_2 \vee \cdots \vee g_n \vee r_2$.

(1) If $r_1 \wedge r_2 \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{h_1 \vee \cdots \vee h_m \vee \boxed{r_1} \qquad g_1 \vee \cdots \vee g_n \vee \boxed{r_2}}{h_1 \vee \cdots \vee h_m \vee \alpha \vee g_1 \vee \cdots \vee g_n \vee \alpha},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_m \vee g_1 \vee \cdots \vee g_n.$$

(2) If $r_1 \wedge g_i \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{\begin{array}{c} h_1 \vee \cdots \vee h_m \vee \boxed{r_1} \\ g_1 \vee \cdots \vee \boxed{g_i} \vee \cdots \vee g_n \vee r_2 \end{array}}{\begin{array}{c} h_1 \vee \cdots \vee h_m \vee \alpha \vee g_1 \vee \cdots \vee \\ g_{i-1} \vee \alpha \vee g_{i+1} \vee \cdots \vee g_n \vee r_2 \end{array}},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_m \vee g_1 \vee \cdots \vee g_{i-1} \vee g_{i+1} \vee \cdots \vee g_n \vee r_2.$$

(3) If $h_i \wedge r_2 \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{\begin{array}{c} h_1 \vee \cdots \vee \boxed{h_i} \vee \cdots \vee h_m \vee r_1 \\ g_1 \vee \cdots \vee g_n \vee \boxed{r_2} \end{array}}{\begin{array}{c} h_1 \vee \cdots \vee h_{i-1} \vee \alpha \vee h_{i+1} \vee \\ \cdots \vee h_m \vee r_1 \vee g_1 \vee \cdots \vee g_n \vee \alpha \end{array}},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_{i-1} \vee h_{i+1} \vee \cdots \vee h_m \vee g_1 \vee \cdots \vee g_n \vee r_1.$$

Case 2: $H_1 = h_1 \vee h_2 \vee \cdots \vee h_m \vee r$, $H_2 = g_1 \vee g_2 \vee \cdots \vee g_n$.

(1) If $r \wedge g_i \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{\begin{array}{c} h_1 \vee \cdots \vee h_m \vee \boxed{\text{r}} \\ g_1 \vee \cdots \vee \boxed{g_i} \vee \cdots \vee g_n \end{array}}{\begin{array}{c} h_1 \vee \cdots \vee h_m \vee \alpha \vee g_1 \vee \cdots \vee \\ g_{i-1} \vee \alpha \vee g_{i+1} \vee \cdots \vee g_n \end{array}},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_m \vee g_1 \vee \cdots \vee g_{i-1} \vee g_{i+1} \vee \cdots \vee g_n.$$

(2) If $h_i \wedge g_j \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{\begin{array}{c} h_1 \vee \cdots \vee \boxed{h_i} \vee \cdots \vee h_m \vee r \\ g_1 \vee \cdots \vee \boxed{g_j} \vee \cdots \vee g_n \end{array}}{\begin{array}{c} h_1 \vee \cdots \vee h_{i-1} \vee \alpha \vee h_{i+1} \vee \cdots \vee g_1 \\ \vee \cdots \vee g_{j-1} \vee \alpha \vee g_{j+1} \vee \cdots \vee g_n \vee r \end{array}},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_{i-1} \vee h_{i+1} \vee \cdots \vee g_1 \vee \cdots \vee g_{j-1} \vee g_{j+1} \vee \cdots \vee g_n \vee r.$$

Case 3: $H_1 = h_1 \vee h_2 \vee \cdots \vee h_m \vee r_1$, $H_2 = r_2$.

(1) If $r_1 \wedge r_2 \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{h_1 \vee \cdots \vee h_m \vee \boxed{r_1} \quad \boxed{r_2}}{h_1 \vee \cdots \vee h_m \vee \alpha \vee \alpha},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee h_2 \vee \cdots \vee h_m.$$

(2) If $h_m \wedge r_2 \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{h_1 \vee h_2 \vee \cdots \vee h_{m-1} \vee \boxed{h_m} \vee r_1 \quad \boxed{r_2}}{h_1 \vee \cdots \vee h_{m-1} \vee \alpha \vee r_1 \vee \alpha},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_{m-1} \vee r_1.$$

Case 4: $H_1 = h_1 \vee h_2 \vee \cdots \vee h_m$, $H_2 = r$.
If $h_m \wedge r \leqslant \alpha$, then the resolution of H_1 and H_2 is represented as

$$\frac{h_1 \vee \cdots \vee \boxed{h_m} \quad \boxed{r}}{h_1 \vee \cdots \vee h_{m-1} \vee \alpha \vee \alpha},$$

and the resolvent of H_1 and H_2 is

$$R_\alpha(H_1, H_2) = \alpha \vee h_1 \vee \cdots \vee h_{m-1}.$$

Where, $h_i (i = 1, 2, \cdots, m)$ and $g_i (i = 1, 2, \cdots, n)$ are the negative g-literals, r, r_1 and r_2 are the positive g-literals.

Remark 4.

(1) Specially, if H_1 is a lattice-valued Horn g-clause, then $\alpha \vee H_1$ is still called lattice-valued Horn g-clause. Here, α is regard as an identification and doesn't implement the resolution.

(2) Obviously, the resolvent of two lattice-valued Horn g-clauses is still lattice-valued Horn g-clause.

(3) There exists no self-resolution in every resolution process.

Example 7. In lattice-valued propositional logic system $\mathscr{L}_6P(X)$, let H_1 and H_2 be lattice-valued g-Horn clauses, and $H_1 = (p \to q)' \vee (r \to a)' \vee p$, $H_2 = (p \to r)' \vee (p \to b)$, $a, b \in L_6$.

According to Definition 9, since $p \wedge (p \to b) \leqslant \alpha$, then we obtain the resolution process

$$\frac{(p \to q)' \vee (r \to a)' \vee \boxed{p} \quad (p \to r)' \vee \boxed{(p \to b)}}{(p \to q)' \vee (r \to a)' \vee \alpha \vee (p \to r)' \vee \alpha},$$

and the resolvent of H_1 and H_2 is as follows:

$$R_\alpha(H_1, H_2) = \alpha \vee (p \to q)' \vee (r \to a)' \vee (p \to r)'.$$

Theorem 3. *In lattice-valued propositional logic system $\mathscr{L}P(X)$, H_1 and H_2 are the lattice-valued Horn g-clauses, and $H_1 = h_1 \vee h_2 \vee \cdots \vee h_m \vee r_1$, $H_2 = g_1 \vee g_2 \vee \cdots \vee g_n \vee r_2$, $\alpha \in L$. If $R_\alpha(H_1, H_2)$ is the resolvent of H_1 and H_2, then $H_1 \wedge H_2 \leqslant R_\alpha(H_1, H_2)$.*

Proof. Let $H_1 = G_1 \vee r_1$, $H_2 = G_2 \vee r_2$. Since $R_\alpha(H_1, H_2)$ is the resolvent of H_1 and H_2, we suppose that there exist r_1 in H_1 and r_2 in H_2 such that $r_1 \wedge r_2 \leqslant \alpha$, then

$$\begin{aligned} H_1 \wedge H_2 &= (G_1 \vee r_1) \wedge (G_2 \vee r_2) \\ &= (G_1 \wedge G_2) \vee (G_1 \wedge r_2) \vee \\ &\quad (r_1 \wedge G_2) \vee (r_1 \wedge r_2) \\ &\leqslant \alpha \vee (G_1 \wedge G_2) \vee (G_1 \wedge r_2) \vee (r_1 \wedge G_2) \\ &\leqslant (\alpha \vee G_1) \vee (\alpha \vee G_2) \\ &= \alpha \wedge G_1 \vee G_2. \end{aligned}$$

Hence, $H_1 \wedge H_2 \leqslant R_\alpha(H_1, H_2)$. □

Definition 19. In lattice-valued propositional logic system $\mathscr{L}P(X)$, let $S = \{H_1, \cdots, H_i, \cdots, H_n\}$ be a set of lattice-valued Horn g-clauses, $\alpha \in L$. $\omega = \{D_1, D_2, \cdots, D_i, \cdots, D_k\}$ is an α-resolution deduction from S to lattice-valued Horn g-clause D_k if it satisfies the following condition
(1)$D_i \in S, i = 1, 2, \cdots, k$; or
(2)there exist m and j, such that $D_i = R_\alpha(D_m, D_j)(m < i, j < i)$.

Theorem 4. *(soundness) In lattice-valued propositional logic system $\mathscr{L}P(X)$, let S be a set of lattice-valued Horn g-clauses $\alpha \in L$, and $\omega = \{D_1, \cdots, D_i, \cdots, D_k\}$ an $\alpha-$resolution deduction from S to lattice-valued Horn g-clause D_k. If D_k is $\alpha - \diamond$, then $S \leqslant \alpha$, i.e., if $D_k = \alpha$, then $S \leqslant \alpha$.*

Proof. Since $\{D_1, \cdots, D_i, \cdots, D_m\}$ is an $\alpha-$resolution deduction from S to lattice-valued Horn g-clause D_m and $D_m = \alpha$, according to Theorem 3, then

$$S \leqslant S \wedge D_1 \wedge \cdots \wedge D_i \wedge \cdots \wedge D_m.$$

Moreover, $D_k = \alpha$, then

$$S \wedge D_1 \wedge \cdots \wedge D_i \wedge \cdots \wedge D_m \leqslant \alpha.$$

i.e., $S \leqslant \alpha$. Therefore, the theorem holds. □

Theorem 5. *(completeness) Let S be a set of lattice-valued Horn g-clauses in lattice-valued propositional logic system $\mathscr{L}P(X)$, $\alpha \in L$. If $S \leqslant \alpha$, then there exists an α-resolution deduction from the set S of lattice-valued Horn clauses to $\alpha - \diamond$.*

Proof. Due to the form of lattice-valued Horn g-clauses, we can regard them as general g-clauses. Moreover, in a resolution process, the resolution is implemented as the rule of Definition 17 in order to obtain the resolvent. Analogous to theorem 11.3.2 in the reference [10], the theorem holds. □

4. An α-resolution algorithm for a set of lattice-valued Horn g-clauses in $\mathscr{L}P(X)$

According to the α-resolution method for lattice-valued Horn g-clauses in lattice-valued propositional logic system $\mathscr{L}P(X)$, we can construct an α-resolution algorithm for designing an applied automated reasoning program.

Let $S = \{H_0, H_1, \cdots, H_i, \cdots, H_m\}$ be a set of lattice-valued Horn g-clauses in $\mathscr{L}P(X)$, G_{H_i} a set of g-literals in H_i, $|G_{H_i}|$ is the number of g-literals. If H_i is a center g-clause, then we denote the corresponding resolution g-clause by C_{H_i}. If g is a g-literal, then we denote the resolution g-literal by h_g such that $g \wedge h \leqslant \alpha$.

An α-resolution algorithm for lattice-valued Horn g-clauses is constructed as follows:

Step 1. set H_0 as a top lattice-valued Horn g-clause such that $S-\{H_0\}$ is α-satisfiable;

Step 2. set $i=0$;

Step 3. take a g-literal g in G_{H_i}, search the lattice-valued Horn g-clause C_{H_i} in S_i such that $h_g \in C_{H_i}$ and $g \wedge h_g \leqslant \alpha$;

Step 4. set C_{H_i} as the resolution g-clause, and obtain the resolvent $D_i = R_\alpha(H_i, C_{H_i})$,

a. if $D_i = \alpha - \diamond$, then the algorithm is terminated, go to Step 6;

b. if $D_i \neq \alpha - \diamond$, then add the resolvent D_i to the set S_i, go to Step 5;

Step 5. $i++$, $H_i = D_i$, $S_i = S_{i-1} \cup \{D_i\}$, go to Step 3;

Step 6. the algorithm is terminated.

Example 8. Let $\mathscr{L} = (L_{9\times 2}, \vee, \wedge, ', \rightarrow, (Ab,F), (Ab,T))$ be lattice implication algebra and $\mathscr{L}_{V(9\times 2)}P(X)$ lattice-valued propositional logic system based on $\mathscr{L}$. Suppose that $S = \{H_1, H_2, H_3, H_4, H_5\}$ is a set of lattice-valued Horn g-clauses, and

$$\begin{aligned} H_1 &= (p \rightarrow q)' \vee (s \rightarrow (So,T)), \\ H_2 &= ((Al,T) \rightarrow p)' \vee (p \rightarrow q), \\ H_3 &= p, \\ H_4 &= (p \rightarrow q)' \vee ((So,T) \rightarrow s), \\ H_5 &= (t \rightarrow (Qu,T))' \vee s, \end{aligned}$$

where p,q,s,t are the literals in $\mathscr{L}_{V(9\times 2)}P(X)$.

According to the α-resolution method, let (Ex,T) be a resolution level α, H_3 the top generalized clause. By using the present algorithm, a resolution deduction process is as follows:

(1) Since $p \wedge ((Al,T) \rightarrow p)' \leqslant \alpha$, then the resolution of H_3 and H_2 is represented as

$$\frac{\boxed{p} \qquad \boxed{((Al,T) \rightarrow p)'} \vee (p \rightarrow q)}{\alpha \vee \alpha \vee (p \rightarrow q)},$$

and the resolvent of H_3 and H_2 is

$$R_\alpha(H_3, H_2) = \alpha \vee (p \rightarrow q).$$

$R_\alpha(H_3, H_2)$ is denoted by D_1.

(2) Since $(p \rightarrow q) \wedge (p \rightarrow q)' \leqslant \alpha$, then the resolution of D_1 and H_1 is represented as

$$\frac{\alpha \vee \boxed{(p \rightarrow q)} \qquad \boxed{(p \rightarrow q)'} \vee (s \rightarrow (So,T))}{\alpha \vee \alpha \vee \alpha \vee (s \rightarrow (So,T))},$$

and the resolvent of D_1 and H_1 is

$$R_\alpha(D_1, H_1) = \alpha \vee (s \rightarrow (So,T)).$$

$R_\alpha(D_1, H_1)$ is denoted by D_2.

(3) Since $(s \rightarrow (So,T)) \wedge s \leqslant \alpha$, then the resolution of D_2 and H_5 is represented as

$$\frac{\alpha \vee \boxed{(s \rightarrow (So,T))} \qquad (t \rightarrow (Qu,T))' \vee \boxed{s}}{\alpha \vee \alpha \vee (t \rightarrow (Qu,T))' \vee \alpha},$$

and the resolvent of D_2 and H_5 is

$$R_\alpha(D_2, H_5) = \alpha \vee (t \rightarrow (Qu,T))'.$$

$R_\alpha(D_2, H_5)$ is denoted by D_3.

(4) Since $(t \rightarrow (Qu,T))' \wedge ((So,T) \rightarrow s) \leqslant \alpha$, then the resolution of D_3 and H_4 is represented as

$$\frac{\alpha \vee \boxed{(t \rightarrow (Qu,T))'} \qquad (p \rightarrow q)' \vee \boxed{((So,T) \rightarrow s)}}{\alpha \vee \alpha \vee (p \rightarrow q)' \vee \alpha},$$

and the resolvent of D_3 and H_4 is

$$R_\alpha(D_3, H_4) = \alpha \vee (p \rightarrow q)'.$$

$R_\alpha(D_3, H_4)$ is denoted by D_4.

(5) Since $(p \rightarrow q)' \wedge (p \rightarrow q) \leqslant \alpha$, then the resolution of D_4 and D_1 is represented as

$$\frac{\alpha \vee \boxed{(p \rightarrow q)'} \qquad \alpha \vee \boxed{(p \rightarrow q)}}{\alpha \vee \alpha \vee \alpha \vee \alpha},$$

and the resolvent of D_4 and H_1 is

$$R_\alpha(D_4, D_1) = \alpha.$$

Hence, we obtain the following formula from the above resolution process.

$$H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge H_5 \wedge D_1 \wedge D_2 \wedge D_3 \wedge D_4 \wedge D_5 \leqslant \alpha.$$

Therefore, $S \leqslant \alpha$, and there exists an α-resolution deduction from S to $\alpha - \diamond$ with the top lattice-valued Horn g-clause H_3.

5. Conclusions

In this paper we establish α-resolution method for lattice-valued Horn generalized clauses and construct an α-resolution algorithm in lattice-valued propositional logic system $\mathscr{L}P(X)$. At the same time, we also give the soundness theorem and completeness theorem. Lattice-valued Horn generalized clause, which is an important type of generalized clauses, can express some special information in real world. In the future work, The practical application of these results to complex systems will be investigated and reported in order to support the reasoning system in intelligent information process. Moreover, α-resolution method for lattice-valued Horn generalized clauses can be studied in lattice-valued first-order logic system $\mathscr{L}F(X)$.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China(Grant No. 61175055, U1304609, 61100046, 61300123), the National High Technology Research and Development Program of China(863 Program, No. 2012AA101608), the High-level Talent Foundation of Henan University of Technology(No. 2012BS012), Key Technology Research Program of the Education Department Henan Province(No. 13B520945), the Plan of Nature Science Fundamental Research in Henan University of Technology (No. 2012JCYJ04), the Fundamental Research Funds for the Henan Provincial Colleges and Universities (No. 2014YWQQ12).

The authors would like to thank the reviewers for their important comments that greatly enhanced the quality of the paper.

References

1. J.A. Robinson, A machine-oriented logic based on the resolution principle, *J. ACM*, **12(1)**, 23-41(1965) .
2. C.L. Chang, R.C.T. Lee, Symbolic Logic and Mechanical Theorem Proving, *Academic Press*, New York(1973).
3. D.W. Loveland, Automated Theorem Proving: A logical Basis, North-Holland, Amsterdam(1978).
4. X.H. Liu, Resolution-based Automated Reasoning, *Academic Press of China*, Beijing, China(1994)(in Chinese).
5. R.Q. Lu, Resolution principle, *Science China* **7**(1981). 896-903 (in Chinese).
6. Y. Xu, Lattice implication algebra, *J. Southwest Jiaotong University*, **28(1)**, 20-27(1993)(in Chinese).
7. Y. Xu, K. Qin, Lattice-valued propositionallogic(I), *J. Southwest Jiaotong University*, **2**, 123-128, (1993)(English version).
8. Y. Xu, K. Qin, Lattice-valued propositional logic(II), *J. Southwest Jiaotong University*, **1**, 22-27(1994)(English version).
9. Y. Xu, D. Ruan, E.E. Kerre, J. Liu, α-Resolution principle based on lattice-valued propositional logic LP(X), *Information Science*, **130(1-4)**, 195-223(2000).
10. Y. Xu, D. Ruan, K. Qin and J. Liu, Lattice-Valued Logic: An alternative approach to treat fuzziness and incomparability, *Springer-Verlag*, Berlin(2003).
11. 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).
12. J. Liu, D. Ruan, Y. Xu, Z.M. Song, A resolution-like strategy based on lattice-valued logic, *IEEE Transactions on Fuzzy Systems*, **11(4)**, 560-567(2003).
13. J. Liu, Z.M. Song, K.Y. Qin, A resolution procedure based on a fuzzy logic, *The Ninth IEEE International Conference on Fuzzy Systems*, **1**, 191-196(2000).
14. J. Liu, M.L. Luis, Y. Xu, Z.R. Lu, Automated Reasoning Algorithm for Linguistic Valued Lukasiewicz Propositional Logic, *Proceedings of the 37th International Symposium on Multiple-Valued Logic(ISMVL'07)*, Oslo, Norway, 13-16 May, (2007).
15. W.T. Xu, W.Q. Zhang, D.X. Zhang, Y. Xu, X.D. Pan, α-resolution method for lattice-valued Horn generalized clauses in lattice-valued propositional logic system, *the 11th International FLINS Conference: Decision Making and Soft Computing*, August 17-20, Brazil, 270-275(2014).
16. X.F. Tang, C.J. Jiang, M.C. Zhou, Automatic Web

service composition based on Horn clauses and Petri nets, *Expert Systems with Applications*, **38**, 13027-13031(2011).
17. Z. Pei, D. Ruan, J. Liu, Y. Xu, Linguistic Values Based Intelligent Information Processing: Theory, Methods, and Applications, *Atlantis Press and World Scientific Publishing Company*(2010).
18. Yang Xu, Shuwei Chen and Jun Ma, Linguistic Truth-valued Lattice Implication Algebra and Its Properties, *in: IMACS Multiconference on Computational Engineering in Systems Applications*, 1413-1418 (2006).