The Generalized Literals Of $L_6P(X)$ With Linguistic Truth-value

W. Wang[1,3] B. Jiang[2] Y. Xu[2]

[1] School of Electric Engineering,Southwest Jiaotong University, Chengdu, 610031, China
[2] Department of Mathematics, Southwest Jiaotong University, Chengdu, 610031, China
[3] Department of Mathematics, PLA Foreign Language University, Luoyang, 471003, China

Abstract

It is only a formalized statement that the research has carried out on inference with linguistic truth-value, and it isn't on a logical basis. An e ective inference system needs a sound and complete logic as it basis. For using lattice-valued propositional logic $LP(X)$ and -resolution principle on $L_6P(X)$, which were given by Xu, to deal with uncertainty information which have linguistic truth-value, in this paper, we discuss the properties of lattice implication algebra L_6 which consist of linguistic terms: $true$, $false$, $more\ true$, $more\ false$, $less\ true$, $less\ false$. The properties of Propositional Logic $L_6P(X)$, and the structures of n generalized literals, $n \leqslant 2$, were obtained.

Keywords: linguistic term; linguistic truth-value; lattice implication algebra; propositional variable; generalized literal

1. Introduction

Since 1990, there have been some important conclusions on inference with linguistic truth-value. In 1990, Ho [1] constructed a distributive lattice-Hedge algebra, which can be used to deal with proposition with linguistic truth-value. In 1996, Zedah [2] discussed the formalization of some words and propositions of natural language, he gave the standardization forms of language propositions and production rulers, and discussed the fuzzy inference, which have linguistic truth-value, based on the fuzzy sets theory and the fuzzy inference method. Between 1998 and 1999, Turksen [4, 5]studied the formalization and the inference of descriptive words, substantive words and declarative sentence.

In 1993, Xu Yang [6] made lattice and implication algebra combined, and found a new algebra-lattice implication algebra. In lattice implication algebra, the implication operation, which is described by axioms, is a generalization of kleene's implication. The lattice-valued propositional logical system $LP(X)$, which is based on lattice implication algebra L, was constructed by Xu Yang [7]. The -resolution principle on $LP(X)$, which is a generalization of the classical resolution principle, was given by Xu Yang [8]. The -resolution principle can be used to prove a set of lattice-valued logical formulas is -false ($\in L$). By using the lattice-valued logic $LP(X)$ to study the uncertainty inference and automated inference which have linguistic truth-value, in 2004, Pei Zheng and Xu Yang [9] gave a lattice implication algebra model of a kind of linguistic variable truth, and discussed the inference on it.

This paper, we gave a lattice implication algebra L_6 which consist of linguistic terms: $true$, $false$, $more\ true$, $more\ false$, $less\ true$, $less\ false$. The properties of L_6, and the properties of $L_6P(X)$ were discussed. It will be a base for the automated inference which deal with uncertainty information with linguistic truth-value.

2. Preliminary

2.1. Implication algebra L_6

We call it a linguistic variable that the variable has one word or one sentence of natural language as its value. For studying the algebraic properties of linguistic terms set which have hedge words, Ho gave a Hedge algebra structure. For using lattice implication algebra and lattice-valued propositional logical system $LP(X)$ to deal with information which have linguistic truth-value, we selected $C = \{true, false\}$, $H = \{more, I, less\}$(I is the identity), and got a lattice implication algebra L_6 as Fig.1.

In L_6, the order-reversing involution "′", which is equivalent to the negative connection of natural language, is de ned as follows: $true' = false$,

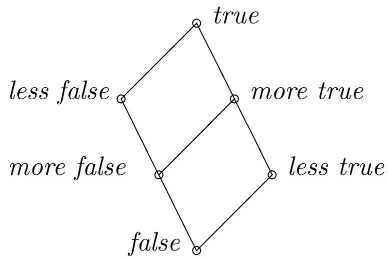

*Fig.*1 Hasse Diagram of L_6.

$false' = true$, $(more\ true)' = (more\ false)'$, $(more\ false)' = (more\ true)'$,$(less\ true)' = (less\ false)'$, $(less\ false)' = (less\ true)'$.

The implication operator "→" , which is equivalent to the implication connection of natural language, is de ned as Table 1. Here, $T = true, F = false, MT = more\ true, MF = more\ false, LT = less\ true, LF = less\ false$.

Table 1. Implication operator "→"

→	F	MF	LF	LT	MT	T
F	T	T	T	T	T	T
MF	MT	T	T	MT	T	T
LF	LT	MT	T	LT	MT	T
LT	LF	LF	LF	T	T	T
MT	MF	LF	LF	MT	T	T
T	F	MF	LF	LT	MT	T

According to the properties of lattice implication algebra, we have the following theorem.

Theorem 1 *For L_6, we have conclusions as following:*

1. *For arbitrary $a \in L_6$, $a \to false = a'$.*
2. *For arbitrary $a \in L_6$,*

$$less\ false \to a = (less\ false)' \vee a,$$

$$less\ true \to a = (less\ true)' \vee a,$$

3. *For arbitrary $a \in L_6$, if $a \neq more\ false$, and $a \neq more\ true$, then $a \vee a' = true$.*

3. Main Results

3.1. Propositional logic $L_6P(X)$

For convenience, we give some basic concepts of $L_6P(X)$.

De nition 1 *Let X be a set of propositional variables, $T = L_6 \cup \{\prime, \to\}$ be a type with $ar(a) = 0$, $ar(\prime) = 1$ and $ar(\to) = 2$ for every $a \in L$. The propositional algebra of the lattice-valued propositional calculus on the set of propositional variables is the free T-algebra on X and denoted by $L_6P(X)$.*

In $L_6P(X)$, we write $(\prime, x)$ as x', $(\to, x, y)$ as $x \to y$, and call the elements of L_6 are constants.

De nition 2 *If mapping $v : L_6P(X) \to L_6$ is a homomorphism of T algebra, we call v a valuation of $L_6P(X)$.*

It is clear that if v is a valuation of $L_6P(X)$, then, for every $a \in L_6$, we have $v(a) = a$. For arbitrary mapping $\varphi : X \to L_6$, there exists a valuation v of $L_6P(X)$, such that $v|_X = \varphi$.

Since $L_6P(X)$ and L_6 are free T algebra and have the same type, $L_6P(X)$ is also a lattice implication algebra.

De nition 3 *F and G are formulae of $L_6P(X)$, if for arbitrary valuation v, $v(F) = v(G)$, we call F and G are equivalence, and denoted as $F = G$.*

Theorem 2 *For every logical formula F, then*

1. *$false \to F = true, true \to F = F$.*
2. *$less\ true \to F = (less\ true)' \vee F$.*
3. *$less\ false \to F = (less\ false)' \vee F$.*

proof It is straightforward.

In lattice implication algebra L_6, *more true* → *more false* and *more false* → *more true* do not act like Kleene implications, and others act like Kleene implications. According to Theorem 2, if $a \neq$ *more false*, $a \neq$ *more true*, then $a \to F$ and $F \to a$ are Kleene implications. Here, F is a logical formula of $L_6P(X)$. It is well know that a logical formula, which only contains Kleene implications, can be translated into a conjunctive (disjunctive) normal form.

3.2. Literals of $L_6P(X)$

A lattice-valued propositional logical formula F of $L_6P(X)$ is called an extremely simple form, for short ESF, if a lattice-valued propositional logical formula F^* obtained by delating any constant or literal or implication term appearing in F is not equivalent to F. A lattice-valued propositional logical formula F of $L_6P(X)$ is called an indecomposable extremely simple form, for short $IESF$, if F is an ESF containing no connective other than implication connectives and order-reversing involutions. An $IESF$ is called an n-$IESF$ if there is n implication connective occurring in it.

De nition 4 *All the IESFs of $L_6P(X)$ are called generalized literals.*

For convenience, we call an n-$IESF$ G an n-generalized literal. In general, the constants and variables of $L_6P(X)$ are called 0-generalized literals, the order-reversing involution of variables of $L_6P(X)$ are called 1-generalized literals. If F is an n-generalized literal($n \geqslant 1$), and F' is also a generalized literal, then F' is a $(n+1)$-generalized literal.

According to De nition 4, a generalized literal isn't equivalent to a conjunctive (disjunctive) normal form of any subformulae of itself. In other words, a generalized literal is an indecomposable logical formula with respect to conjunction and disjunction, and it is a basic logical formula.

We let $B = \{\textit{more true}, \textit{more false}\}$, $A = \{\textit{more true}, \textit{more false}, \textit{false}\}$.

Theorem 3 *For every n-generalized literal F, $n \geqslant 1$, there is not constant occurring in F, or only the constant of A is occurring in F.*

Proof If $n = 1$, and there exists constant c occurring in F, then F has the following forms: $c \rightarrow x$, $x \rightarrow c$. Here, x is a propositional variable. If $c \notin A$, F is equivalent to the formula of following: x, $true$, $c' \vee x$ and $x' \vee c$. This contradicts that F is a 1-generalized literal.

Suppose the conclusion holds for $n \leqslant k$. If $n = k+1$, according to the Lemma 11.3.2 [10], there exist k_1 generalized F_1 and k_2 generalized literal F_2, such that $F = F_1 \rightarrow F_2$, $k_1 + k_2 + 1 = k + 1$. If there is a constant c occurring in F, then c occurs in F_1 or F_2. If c occurs in F_1, and $k_1 = 0$, then $F = c \rightarrow F_2$, $c \notin A$, and then F is the one of the following forms: $F = F_2$, $F = c' \vee F_2$. This contradicts that F is an n generalized literal. if c occurs in F_1, and $k_1 > 0$, by the proof about $n = 1$ and the hypothesis of induction, we have $c \in A$. If c occurs in F_2, we can prove that the conclusion holds similarly.

According to the mathematics induction, the conclusion holds for every number n.

Theorem 4 *For every $a, b \in B$, we have*

1. $(a \rightarrow b) \rightarrow a = a$;
2. $b \rightarrow (b \rightarrow a) = b \rightarrow a$.

Proof We only give the proof of conclusion (1).

If $a = b$, it is clear that conclusion holds.

If $a \neq b$ and $a = \textit{more false}$, then $(a \rightarrow b) \rightarrow a = true \rightarrow \textit{more false} = a$, that is to say the conclusion holds.

If $a \neq b$ and $a = \textit{more true}$, then $(a \rightarrow b) \rightarrow a = \textit{more true} = a$. This completes the proof.

Theorem 5 *For arbitrary logical formula F and $a, b \in B$, we have*

1. $(a \rightarrow F) \rightarrow (F \rightarrow b) = F \rightarrow b$.
2. $(a \rightarrow F) \rightarrow b = F' \vee b$.
3. $(F \rightarrow b) \rightarrow F = F$.
4. $F \rightarrow (F \rightarrow b) = F \rightarrow b$.

ProofWe only give the proof of conclusion (1), others can be given similarly.

If $a = b$, we know that the conclusion (1) holds by properties of lattice implication algebra.

If $a \neq b$, for arbitrary valuation v of $L_6P(X)$, if $v(F) \in B$, we can prove

$$(a \rightarrow v(F)) \rightarrow (v(F) \rightarrow b) = v(F) \rightarrow b$$

easily; if $v(F) \notin B$,

$$(a \rightarrow v(F)) \rightarrow (v(F) \rightarrow b) = (v(F) \vee a') \rightarrow (v(F)' \vee b)$$

$$= (v(F)' \vee b) \wedge true = v(F)' \vee b = v(F) \rightarrow b.$$

Hence, $(a \rightarrow F) \rightarrow (F \rightarrow b) = F \rightarrow b$ holds by the arbitrarily of valuation v.

De nition 5 *We call a logical formula F a implicative formula, if F only contains non-Kleene's implicative connection.*

Theorem 6 *For every logical formula F of $L_6P(X)$, there exist implicative formulae F_{jk}, $j \in J$, $k \in K$, where J, K are nite indexing sets, such that $F = \bigvee\limits_{j \in J} \bigwedge\limits_{k \in K} F_{jk}$.*

Proof If F is an implicative formula, it is clear that the conclusion holds.

If F has 1 Kleene's implicative connective, then F is one of the following forms: $a \rightarrow x$, $x \rightarrow a$. Here, $a \neq \textit{more true}$, and $a \neq \textit{more false}$. We know that the conclusion holds by Theorem 1.

Suppose that the conclusion holds when F contains k implicative connectives. If F contains $k+1$ implicative connectives, and $F = a \rightarrow F_1$, where this implicative connective is Kleene's implicative connective, then $a \rightarrow F_1 = a' \vee F_1$ by Theorem 1, and F can be translated into a conjunction of subformulae F_1^* and F_2^* which have a' and F_1 as its subformulae respectively, by Theorem 2. Since the numbers of implicative connectives occurring in F_1^* and F_2^* are all less than k, according to the hypothesis of induction, there exist logical formulaes F_{jk}^1,

$j \in J_1$, $k \in K_1$, and F^2_{jk}, $j \in J_2$, $k \in K_2$, J_1, K_1, where J_2, K_2 are nite indexing sets, such that

$$F_1^* = \bigvee_{j \in J_1} \bigwedge_{k \in K_1} F^1_{jk}, F_2^* = \bigvee_{j \in J_2} \bigwedge_{k \in K_2} F^2_{jk}.$$

Let $J = J_1 \cup J_2$, $K = K_1 \cup K_2$,

$$F_{jk} = \begin{cases} F^1_{jk}, & j \in J_1, k \in K_1; \\ true, & j \in J_1, k \in K_2; \\ true, & j \in J_2, k \in K_1; \\ F^2_{jk}, & j \in J_2, k \in K_2. \end{cases}$$

It is clear that $F = \bigvee_{j \in J} \bigwedge_{k \in K} F_{jk}$, and the conclusion holds.

According to the mathematical induction, the conclusion holds for arbitrary logical formula F.

According to Theorems 4, 5 and 6, we can get the theorems as following.

Theorem 7 *The* 0*-generalized literals,* 1*-generalized literals and* 2*-generalized literals of* $L_6P(X)$ *are as following:* c, x, $x \to a$, $b \to x$, $x \to y$, $(x \to a) \to b$, $a \neq b$, $(b \to x) \to b$, $(b \to x)'$, $(x \to b)'$, $(x \to b) \to y$, $(b \to x) \to y$, $(x \to y) \to a$, $(x \to y) \to z$, $(x \to y) \to x$, $x \to (y \to z)$, $x \to (x \to z)$, $x \to (y \to a)$, $x \to x'$, $b \to (b \to x)$, $b \to (x \to y)$.

Here, $c \in L_6$, $a \in A$, $b \in B$, $x, y, z \in X$.

4. Conclusion

This paper discussed the properties of L_6 and $L_6P(X)$, and gave the structure of generalized literals of $L_6P(X)$. The logic $L_6P(X)$ provided a logical foundation for uncertainty inference including incompletely comparable information with truth-value: *true*, *more true*, *less true*, *false*, *more false*, or *less false*. The work of this paper also provided a foundation for the further study, we could select the resolution level $\in L_6$, study if two generalized literals of $L_6P(X)$ are -resolvable, prove the soundness theorem and the completeness theorem about this -resolution principle on $L_6P(X)$, and give the automated reasoning method based on this -resolution principle, etc.

Acknowledgement

The presented work is supported by the National Natural Science Foundation of China with granted no. 60474022. And the Henan Province Natural Science Foundation of P.R. China (No. 2000520025, G2002026). We gratefully acknowledge the valuable and helpful suggestions given by the anonymous reviewers.

References

[1] Ho C.Nguyen and Wechler, W., Hedge algebras: an algebraic approach to structure of sets of linguistic true values, *Fuzzy set and systems*, pp. 281-293, 35(1990).

[2] L.A.Zadeh, Reference,"Fuzzy logic = computing with words", *IEEE Trans. Fuzzy systems*, pp. 103-111, 4(1996).

[3] Ho C.Nguyen and Wechler, W., Extended hedge algebras and their application to fuzzy logic, *Fuzzy set and systems*, pp. 259-281, 52(1992).

[4] I.B.Turksen, A.Kandel and Y.Q.Zhang, Universal truth tables and normal forms, *IEEE Fuzzy Syst*, pp. 295-303, 6(2)(1998).

[5] I.B.Turksen, Type I and type II fuzzy system modeling, *Fuzzy set and systems*, pp. 11-34, 106(1999).

[6] Y. Xu, Lattice implication algebra, *Journal of south west Jiaotong Unversity*, pp. 20-27, 28(1)(1993).

[7] K.Y. Qin and Y. Xu, Lattice-valued propositional logic (I), *Journal of south west Jiaotong Unversity (in English)*, 1993, 2: 123-128

[8] Yang Xu, Da Ruan, E.E.Kerre and J.Liu, Resolution principle based on lattice-valued logic $LP(X)$, *Information Science*, pp. 195-223, 130(2000).

[9] Zheng Pei, Yang Xu, Lattice implication algebra model of a kind of linguistic terms and its inference, *FLINS 2004 6th International Conference on Applied Computational Intelligence*, Blankenberghe, Belgium, pp. 93-98, 2004.

[10] Yang Xu, Da Ruan, Keyun Qin, Jun Liu, Lattice-Valued Logic, *Springer*, 2003.

[11] Qin Keyue and Xu Yang, Lattice-valued propositional logic (II), *Journal of south west Jiaotong Unversity (in English)*, 1994, 2: 22-27

[12] Jun Liu, Yang Xu and D.Ruan, Automated Method Based on $LP(X)$, *Proceedings of East West Fuzzy Colloquium 2000 and* 8^{th}, Germany, pp. 60-67, 2000.

[13] Wang Wei, Jiang Baoqing and Xu Yang, Automated Reasoning Method Based on LP(X), *FLINS 2004 6th International Conference on Applied Computational Intelligence,* Blankenberghe, Belgium, pp. 105-110, 2004.