Resolution in Linguistic Propositional Logic based on Linear Symmetrical Hedge Algebra

Thi-Minh-Tam Nguyen¹, Viet-Trung Vu², The-Vinh Doan², and Duc-Khanh Tran²

 ¹ Faculty of Information Technology - Vinh University Email: nmtam@vinhuni.edu.vn
² School of Information and Communication Technology Hanoi University of Science and Technology Email: trungvv91@gmail.com doanthevinh1991@gmail.com khanhtd@soict.hut.edu.vn

Abstract. The paper introduces a propositional linguistic logic that serves as the basis for automated uncertain reasoning with linguistic information. First, we build a linguistic logic system with truth value domain based on a linear symmetrical hedge algebra. Then, we consider Gödel's t-norm and t-conorm to define the logical connectives for our logic. Next, we present a resolution inference rule, in which two clauses having contradictory linguistic truth values can be resolved. We also give the concept of reliability in order to capture the approximative nature of the resolution inference rule. Finally, we propose a resolution procedure with the maximal reliability.

Keywords: Linear Hedge Algebra, Linguistic Truth Value, Linguistic Propositional Logic, Resolution.

1 Introduction

Automated reasoning is an approach to model human thinking. The resolution rule introduced by Robinson (1965) [5] marked an important point in studying automated reasoning. Resolution based on fuzzy set theory of Zadeh [18] has been studied to deal with uncertain information. In fuzzy logic, each clause has a membership function in [0, 1]. Since then subtantial works [5, 17, 1, 13, 6, 14] have been done on the fuzzy resolution.

In two-valued logic, each clause has a truth value True or False. Therefore, the logical inference is absolutely accurate. However, in linguistic logic, each literal has a linguistic truth value such as MoreTrue, MoreFalse, PossibleVeryTrue, LessTrue, ..., where True, False are generators and More, PossibleVery, Less, ... are strings of hedges which increase or decrease the semantic of generators. Thus the accuracy of logical inference is approximate. For instance the two clauses $A^{\text{True}} \vee B^{\text{MoreTrue}}$ and $B^{\text{LessFalse}} \vee C^{\text{True}}$ can be resolved to obtain $A^{\text{True}} \vee C^{\text{True}}$. However the literals B^{MoreTrue} and $B^{\text{LessFalse}}$ are not totally contradictory, they

are only contradictory at a certain degree. Consequently the resolution inference is only reliable at a certain degree. Therefore, when the inference is performed, the infered formula should be associated with a certain reliability. Automated reasoning in linguistic logic has been attracting many researchers. Many works presented resolution algorithms in linguistic logics with truth value domain based on the implication lattice algebraic structures [2,3,15,16,19] or based on hedge algebra [4,8,10,11].

Along the line of these research directions, we study automated reasoning based on resolution for linguistic propositional logic with truth value domain is taken from linear symmetrical hedge algebra. The syntax of linguistic propositional logic is constructed. To define the semantics of logical connectives we consider t-norm and t-conorm operators in fuzzy logic, specially t-norm and tconorm operators of Gödel and Lukasiewicz. We show that logical connectives based on Gödel connectives are more appropriate to construct logical connectives for our linguistic logic. A resolution rule and resolution procedure are given. The concept of reliability of inference is introduced in such a way that the reliability of the conclusion is smaller than or equal to the reliabilities of the premises. We also present a resolution procedure with maximal reliability and prove the soundness and completeness of the resolution procedure.

The paper is structured as follows: Section 2 introduces basic notions and results on linear symmetrical hedge algebras. Section 3 presents the syntax and semantics of our linguistic propositional logic with truth value domain based on linear symmetrical hedge algebra. The resolution rule and resolution procedure are introduced in Section 4. Section 5 concludes and draws possible future work. Proofs of theorems, lemmas and proposition are presented in the Appendix.

2 Preliminaries

We recall only the most important definitions of hedge algebra for our work and refer the reader to [9,7,8] for further details.

We will be working with a class of abstract algebras of the form $AX = (X, G, H, \leq)$ where X is a term set, G is a set of generators, H is a set of linguistic hedges or modifiers, and \leq is a partial order on X. AX is called a *hedge algebra* (HA) if it satisfies the following:

- Each hedge h is said to be positive w.r.t k, i.e. either $kx \ge x$ implies $hkx \ge kx$ or $kx \le x$ implies $hkx \le kx$; similarly h is said to be negative w.r.t k, i.e. either $kx \ge x$ implies $hkx \le kx$ or $kx \le x$ implies $hkx \ge kx$ (for $x \in X$);
- If terms u and v are independent, then, for all $x \in H(u)$, we have $x \notin H(v)$. If u and v are incomparable, i.e. $u \not< v$ and $v \not< u$, then so are x and y, for every $x \in H(u)$ and $y \in H(v)$;
- If $x \neq hx$, then $x \notin H(hx)$, and if $h \neq k$ and $hx \leq kx$, then $h'hx \leq k'kx$, for all $h, k, h', k' \in H$ and $x \in X$. Moreover, if $hx \neq kx$, then hx and kx are independent;
- If $u \notin H(v)$ and $u \leq v(u \geq v)$, then $u \leq hv(u \geq hv)$ for any $h \in H$.

Let $AX = (X, G, H, \leq)$ where the set of generators G contains exactly two comparable ones, denoted by $c^- < c^+$. For the variable Truth, we have $c^+ =$ $True > c^- = False$. Such HAs are called symmetrical ones. For symmetrical HAs, the set of hedges H is decomposed into two disjoint subsets $H^+ = \{h \in$ $H|hc^+ > c^+\}$ and $H^- = \{h \in H|hc^+ < c^+\}$. Two hedges h and k are said to be converse if $\forall x \in X, hx \leq x$ iff $kx \geq x$, i.e., they are in different subsets; h and k are said to be compatible if $\forall x \in X, hx \leq x$ iff $kx \leq x$, i.e. they are in the same subset. Two hedges in each of the sets H^+ and H^- may be comparable. Thus, H^+ and H^- become posets.

A symmetrical HA $AX = (X, G = \{c^-, c^+\}, H, \leq)$ is called a *linear symmetrical HA* (LSHA, for short) if the set of hedges H can be decomposed into $H^+ = \{h \in H | hc^+ > c^+\}$ and $H^- = \{h \in H | hc^+ < c^+\}$, and H^+ and H^- are linearly ordered.

Let $h_n h_{n-1} \ldots h_1 u$, $k_m k_{m-1} \ldots k_1 u$ be the canonical presentations of values x, y respectively. x = y iff m = n and $h_j = k_j$ for every $j \leq n$. If $x \neq y$ then there exists an $j \leq min\{m, n\} + 1$ (there is one convention is understood that if $j = min\{m, n\} + 1$, then $h_j = I$ where $j = n + 1 \leq m$ or $k_j = I$ where $j = m + 1 \leq n$) such that $h_{j'} = k_{j'}$ with all j' < j. Denote that $x_{<j} = h_{j-1}h_{j-2}\ldots h_1 u = k_{j-1}k_{j-2}\ldots k_1 u$, we have: x < y iff $h_j x_{<j} < k_j x_{<j}, x > y$ iff $h_j x_{<j} > k_j x_{<j}$.

Let x be an element of the hedge algebra AX and the canonical representation of x is $x = h_n \dots h_1 a$ where $a \in \{c^+, c^-\}$. The contradictory element of x is an element y such that $y = h_n \dots h_1 a'$ where $a' \in \{c^+, c^-\}$ and $a' \neq a$, denoted by \overline{x} . In LSHA, every element $x \in X$ has a unique contradictory element in X.

It is useful to limit the set of values X only consists of finite length elements. This is entirely suitable with the practical application in natural language, which does not consider infinite number of hedge of string.

From now on, we consider a LSHA $AX = (X, G, H, \leq, \neg, \lor, \land, \rightarrow)$ where $G = \{\bot, \mathsf{False}, \mathsf{W}, \mathsf{True}, \top\}; \bot, \top$ and W are the least, the neutral, and the greatest elements of X, respectively; $\bot < \mathsf{False} < \mathsf{W} < \mathsf{True} < \top$.

3 Propositional logic with truth value domain based on symmetrical linear hedge algebra

Below, we define the syntax and semantics of the linguistic propositional logic.

Definition 31 An alphabet consists of:

- constant symbols: MoreTrue, VeryFalse, \bot , \top , ...
- propositional variables: A, B, C, \dots
- logical connectives: $\lor, \land, \rightarrow, \neg, \equiv$, and
- auxiliary symbols: \Box , (,), ...

Definition 32 An atom is either a propositional variable or a constant symbol.

Definition 33 Let A be an atom and α be a constant symbol. Then A^{α} is called a literal.

Definition 34 Formulae are defined recursively as follows:

- either a literal or a constant is a formula,
- if P is a formula, then $\neg P$ is a formula, and
- if P, Q are formulae, then $P \lor Q, P \land Q, P \to Q, P \leftrightarrow Q$ are formulae.

Definition 35 A clause is a finite disjunction of literals, which is written as $l_1 \vee l_2 \vee ... \vee l_n$, where l_i is a literal. An empty clause is denoted by \Box .

Definition 36 A formula F is said to be in conjunctive normal form (CNF) if it is a conjunction of clauses.

In many-valued logic, sets of connectives called Lukasiewicz, Gödel, and product logic ones are often used. Each of the sets has a pair of residual t-norm and implicator. However, we cannot use the product logic connectives when our truth values are linguistic.

We recall the operators t-norm(T) and t-conorm(S) on fuzzy logic. It is presented detailed in [12,14].

T-norm is a dyadic operator on the interval $[0, 1]: T : [0, 1]^2 \longrightarrow [0, 1]$, satisfying the following conditions:

- Commutativity: T(x, y) = T(y, x),
- Associativity: T(x, T(y, z)) = T(T(x, y), z),
- Monotonicity: $T(x, y) \leq T(x, z)$ where $y \leq z$, and
- Boundary condition: T(x, 1) = x.

If T is a t-norm, then its dual t-conorm S is given by S(x, y) = 1 - T(1 - x, 1 - y). Let $K = \{n | n \in \mathbb{N}, n \leq N_0\}$. Extended T-norm is a dyadic operator T_E :

- $K^2 \longrightarrow K$ and satisfies the following conditions:
- Commutativity: $T_E(m, n) = T_E(n, m)$,
- Associativity: $T_E(m, T_E(n, p)) = T_E(T_E(m, n), p),$
- Monotonicity: $T_E(m, n) \leq T_E(m, p)$ where $n \leq p$, and
- Boundary condition: $T_E(n, N_0) = n$.

The Extended T-conorm is given by: $S_E(m,n) = N_0 - T_E(N_0 - n, N_0 - m)$. It is easy to prove that S_E is commutative, associate, monotonous. The boundary condition of S_E is: $S_E(0,n) = n$.

Two common pairs (T, S) in fuzzy logic: Gödel's(T, S) and Łukasiewicz's(T, S) are defined as following:

– Gödel:

- $T_G(m,n) = \min(m,n)$
- $S_G(m,n) = \max(m,n)$
- Łukasiewicz:
 - $T_L(m,n) = \max(0, m+n-N_0)$
 - $S_L(m, n) = \min(m + n, N_0)$

Given a SLHA AX, since all the values in AX are linearly ordered, we assume that they are $\perp = v_0 \leq v_1 \leq v_2 \leq \ldots \leq v_n = \top$.

Clearly, the pair (T, S) is determined only depending on max and min operators. Commonly, the truth functions for conjunctions and disjunctions are t-norms and t-conorms respectively.

Example 31 Consider a SLHA $AX = (X, \{\text{True}, \text{False}\}, \{\text{More}, \text{Less}\}, \leq)$ with Less < More. We assume the length of hedge string is limited at 1. Then $AX = \{v_0 = \text{MoreFalse}, v_1 = \text{False}, v_2 = \text{LessFalse}, v_3 = \text{LessTrue}, v_4 = \text{True}, v_5 = \text{MoreTrue}\}$. We determine the truth value of logical connectives based on t-norm and t-conorm operators of Gödel and Lukasiewicz:

- Gödel:
 - LessFalse \lor False = max{LessFalse, False} = LessFalse
 - MoreTrue \land True = min{MoreTrue, True} = True
- Łukasiewicz:
 - LessFalse \lor False = LessTrue
 - MoreTrue \land True = LessFalse

In fact, if the same clause has two truth values LessFalse or False, then it should get the value LessFalse. In the case of two truth values are MoreTrue and True then it should get the value True. We can see that the logical connectives based on Gödel's t-norm and t-conorm operators are more suitable in the resolution framework than those based on Lukasiewicz's. In this paper we will define logical connectives using Gödel's t-norm and t-conorm operators.

Definition 37 Let S be a linguistic truth domain, which is a SLHA $AX = (X, G, H, \leq)$, where $G = \{\top, \mathsf{True}, \mathsf{W}, \mathsf{False}, \bot\}$. The logical connectives \land (respectively \lor) over the set S are defined to be Gödel's t-norm (respectively t-conorm), and furthermore to satisfy the following:

 $- \neg \alpha = \overline{\alpha}.$ $- \alpha \to \beta = (\neg \alpha) \lor \beta.$

where $\alpha, \beta \in S$.

Proposition 31 Let S be a linguistic truth domain, which is a SLHA $AX = (X, \{\top, \mathsf{True}, \mathsf{W}, \mathsf{False}, \bot\}, H, \leq); \alpha, \beta, \gamma \in X$, we have:

- Double negation: • $\neg \neg \alpha = \alpha$
- Commutative:
 - $\alpha \wedge \beta = \beta \wedge \alpha$
 - $\alpha \lor \beta = \beta \lor \alpha$ • $\alpha \lor \beta = \beta \lor \alpha$
- Associative:
 - $(\alpha \land \beta) \land \gamma = \alpha \land (\beta \land \gamma)$
 - $(\alpha \lor \beta) \lor \gamma = \alpha \lor (\beta \lor \gamma)$ • $(\alpha \lor \beta) \lor \gamma = \alpha \lor (\beta \lor \gamma)$
 - \mathbf{D}
- Distributive:

- $\alpha \land (\beta \lor \gamma) = (\alpha \land \beta) \lor (\alpha \land \gamma)$
- $\alpha \lor (\beta \land \gamma) = (\alpha \lor \beta) \land (\alpha \lor \gamma)$

Definition 38 An interpretation consists of the followings:

- a linguistic truth domain, which is a SLHA $AX = (X, G, H, \leq)$, where the set of generators $G = \{\top, \mathsf{True}, \mathsf{W}, \mathsf{False}, \bot\}$,
- for each constant in the alphabet, the assignment of an element in X,
- for each formula, the assignment of a mapping from X to X.

Definition 39 Let I be an interpretation and A be an atom such that $I(A) = \alpha_1$. Then the truth value of a literal A^{α_2} under the interpretation I is determined uniquely as follows:

 $\begin{array}{l} - \ I(A^{\alpha_2}) = \alpha_1 \wedge \alpha_2 \ if \ \alpha_1, \alpha_2 > \mathsf{W}, \\ - \ I(A^{\alpha_2}) = \neg(\alpha_1 \vee \alpha_2) \ if \ \alpha_1, \alpha_2 \leq \mathsf{W}, \\ - \ I(A^{\alpha_2}) = (\neg \alpha_1) \vee \alpha_2 \ if \ \alpha_1 > \mathsf{W}, \alpha_2 \leq \mathsf{W}, \ and \\ - \ I(A^{\alpha_2}) = \alpha_1 \vee (\neg \alpha_2) \ if \ \alpha_1 \leq \mathsf{W}, \alpha_2 > \mathsf{W}. \end{array}$

Definition 310 The truth value of formulae under an interpretation is determined recursively as follows:

 $\begin{array}{l} - \ I(P \lor Q) = I(P) \lor I(Q), \\ - \ I(P \land Q) = I(P) \land I(Q), \\ - \ I(\neg P) = \neg I(P), \\ - \ I(P \to Q) = I(P) \to I(Q) \end{array}$

The following result follows from the properties of the \wedge and \vee operators.

Proposition 32 Let A, B and C are formulae, and I be an arbitrary interpretation. Then,

- Commutative:
 - $I(A \lor B) = I(B \lor A)$
 - $I(A \wedge B) = I(B \wedge A)$
- Associative:
 - $I((A \lor B) \lor C) = I(A \lor (B \lor C))$
 - $I((A \land B) \land C) = I(A \land (B \land C))$
- Distributive:
 - $I(A \lor (B \land C)) = I((A \lor B) \land (A \lor C))$
 - $I(A \land (B \lor C)) = I((A \land B) \lor (A \land C))$

Definition 311 Let F be a formula and I be an interpretation. Then

- F is said to be true under interpretation I iff $I(F) \ge W$, I is also said to satisfy formula F, F is said to be satisfiable iff there is an interpretation I such that I satisfies F, F is said to be tautology iff it is satisfied by all interpretations; - F is said to be false under interpretation I iff $I(F) \leq W$, I is also said to falsify formula F, F is said to be unsatisfiable iff it is falsified by all interpretations.

Definition 312 Formula B is said to be a logical consequence of formula A, denoted by $A \models B$, if for all interpretation I, I(A) > W implies that I(B) > W.

Proposition 33 Let A and B be formulae. Then, $A \models B$ iff $\models (A \rightarrow B)$.

Definition 313 Two formulae A and B are logically equivalent, denoted by $A \equiv B$, if and only if $A \models B$ and $B \models A$.

Proposition 34 Let A, B and C be formulae. Then the following properties hold:

- Idempotency:
 - $A \lor A \equiv A$
 - $A \wedge A \equiv A$
- Implication:
 - $A \to B \equiv (\neg A) \lor B$
 - $(A \equiv B) \equiv (A \rightarrow B) \land (B \rightarrow A)$
- Double negation:
 - $\neg \neg A \equiv A$
- De Morgan:
 - $\neg (A \lor B) \equiv (\neg A) \land (\neg B)$
 - $\neg (A \land B) \equiv (\neg A) \lor (\neg B)$
- Commutativity:
 - $A \lor B \equiv B \lor A$
 - $A \wedge B \equiv B \wedge A$
- Associativity:
 - $A \lor (B \lor C) \equiv (A \lor B) \lor C$
 - $A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$
- Distributivity:
 - $A \lor (B \land C) \equiv (A \lor B) \land (A \lor C)$
 - $A \land (B \lor C) \equiv (A \land B) \lor (A \land C)$

We will be working with resolution as the inference system of our logic. Therefore formulae need to be converted into conjunctive normal form. The equivalence properties in Proposition 34 ensure that the transformation is always feasible.

Theorem 1. Let F be a formula of arbitrary form. Then F can be converted into an equivalent formula in conjunctive normal form.

4 Resolution

In the previous section, we have described the syntax and semantics of our linguistic logic. In this section, we present the resolution inference rule and the resolution procedure for our logic.

Definition 41 The clause C with reliability α is the pair (C, α) where C is a clause and α is an element of SLHA AX such that $\alpha > W$. The same clauses with different reliabilities are called variants. That is (C, α) and (C, α') are called variants of each other.

For a set of *n* clauses $S = \{C_1, C_2, ..., C_n\}$, where each C_i has a reliability α_i , then the reliability α of *S* is defined as: $\alpha = \alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n$.

An inference rule R with the reliability α is represented as:

$$\frac{(C_1,\alpha_1),(C_2,\alpha_2),\ldots,(C_n,\alpha_n)}{(C,\alpha)}$$

We call α the reliability of R, provided that $\alpha \leq \alpha_i$ for i = 1..n.

Definition 42 The fuzzy linguistic resolution rule is defined as follows:

$$\frac{(A^a \vee B^{b_1}, \alpha_1), (B^{b_2} \vee C^c, \alpha_2)}{(A^a \vee C^c, \alpha_3)}$$

where b_1, b_2 and α_3 satisfy the following conditions:

$$\begin{cases} b_1 \wedge b_2 \leq \mathsf{W}, \\ b_1 \vee b_2 > \mathsf{W}, \\ \alpha_3 = f(\alpha_1, \alpha_2, b_1, b_2) \end{cases}$$

with f is a function ensuring that $\alpha_3 \leq \alpha_1$ and $\alpha_3 \leq \alpha_2$.

 α_3 is defined so as to be smaller or equal to both α_1 and α_2 . In fact, the obtained clause is less reliable than original clauses. The function f is defined as following:

$$\alpha_3 = f(\alpha_1, \alpha_2, b_1, b_2) = \alpha_1 \wedge \alpha_2 \wedge (\neg (b_1 \wedge b_2)) \wedge (b_1 \vee b_2) \tag{1}$$

Obviously, $\alpha_1, \alpha_2 \geq W$, and α_3 depends on b_1, b_2 . Additionally, $b_1 \wedge b_2 \leq W$ implies $\neg(b_1 \wedge b_2) > W$. Moreover, $(b_1 \vee b_2) > W$. Then, by Formula (1), we have $\alpha_3 > W$.

Lemma 1. The fuzzy linguistic resolution rule 42 is sound.

We define a fuzzy linguistic resolution *derivation* as a sequence of the form S_0, \ldots, S_i, \ldots , where:

- each S_i is a set of clauses with a reliability, and

- S_{i+1} is obtained by adding the conclusion of a fuzzy linguistic resolution inference with premises from S_i , that is $S_{i+1} = S_i \cup \{(C, \alpha)\}$, where (C, α) is the conclusion of the fuzzy linguistic resolution

$$\frac{(C_1,\alpha_1),(C_2,\alpha_2)}{(C,\alpha)},$$

and $(C_1, \alpha_1), (C_2, \alpha_2) \in S_i$.

A resolution proof of a clause C from a set of clauses S consists of repeated application of the resolution rule to derive the clause C from the set S. If C is the empty clause then the proof is called a *resolution refutation*. We will represent resolution proofs as *resolution trees*. Each tree node is labeled with a clause. There must be a single node that has no child node, labeled with the conclusion clause, we call it the root node. All nodes with no parent node are labeled with clauses from the initial set S. All other nodes must have two parents and are labeled with a clause C such that

$$\frac{C_1, C_2}{C}$$

where C_1, C_2 are the labels of the two parent nodes. If RT is a resolution tree representing the proof of a clause with reliability (C, α) , then we say that RT has the reliability α .

Different resolution proofs may give the same the conclusion clause with different reliabilities. The following example illustrate this.

Example 41 Let $AX = (X, G, H, \leq, \neg, \lor, \land, \rightarrow)$ be a SRHA where $G = \{\bot, \mathsf{False}, \mathsf{W}, \mathsf{True}, \top\}, \bot, \mathsf{W}, \top$ are the smallest, neutral, biggest elements respectively, and $\bot < \mathsf{False} < \mathsf{W} < \mathsf{True} < \top; H^+ = \{\mathsf{V},\mathsf{M}\}$ and $H^- = \{\mathsf{P},\mathsf{L}\}$ (V = Very, M = More, P = Possible, L = Less); Consider the following set of clauses:

- 1. $A^{\mathsf{MFalse}} \lor B^{\mathsf{False}} \lor C^{\mathsf{VMTrue}}$
- 2. $B^{\mathsf{LTrue}} \lor C^{\mathsf{PTrue}}$
- 3. A^{PTrue}
- 4. B^{VTrue}
- 5. C^{VFalse}

At the beginning, each clause is assigned to the highest reliability \top . We have:

$$\frac{(A^{\mathsf{PTrue}},\top)(A^{\mathsf{MFalse}} \vee B^{\mathsf{False}} \vee C^{\mathsf{VMTrue}},\top)}{(B^{\mathsf{False}} \vee C^{\mathsf{VMTrue}},\mathsf{PTrue})(B^{\mathsf{VTrue}},\top)} \\ \frac{(C^{\mathsf{VMTrue}},\mathsf{PTrue})}{(\Box,\mathsf{PTrue})}$$

$$\frac{(A^{\mathsf{MFalse}} \lor B^{\mathsf{False}} \lor C^{\mathsf{VMTrue}}, \top)(B^{\mathsf{LTrue}} \lor C^{\mathsf{PTrue}}, \top)}{(A^{\mathsf{MFalse}} \lor C^{\mathsf{VMTrue}}, \mathsf{LTrue}) \quad (A^{\mathsf{PTrue}}, \top)} \\ \underbrace{(C^{\mathsf{VMTrue}}, \mathsf{LTrue})}_{(\Box, \mathsf{LTrue})} \quad (C^{\mathsf{VFalse}}, \top) \\ \end{array}$$

Since different proofs of the same clause may have different reliabilities, it is natural to study how to design a resolution procedure with the best reliability. Below we present such a procedure.

We say that a set of clauses S is *saturated* iff for every fuzzy linguistic resolution inference with premises in S, the conclusion of this inference is a variant with smaller or equal reliability of some clauses in S. That is for every fuzzy linguistic resolution inference

$$\frac{(C_1,\alpha_1),(C_2,\alpha_2)}{(C,\alpha)}$$

where $(C_1, \alpha_1), (C_2, \alpha_2) \in S$, there is some clause $(C, \alpha') \in S$ such that $\alpha \leq \alpha'$.

We introduce a resolution strategy, called α -strategy, which guarantees that the resolution proof of each clause has the maximal reliability. An α -strategy derivation is a sequence of the form S_0, \ldots, S_i, \ldots , where

- each S_i is a set of clauses with reliability, and
- S_{i+1} is obtained by adding the conclusion of a fuzzy linguistic resolution inference with premises with maximal reliabilities from S_i , that is $S_{i+1} = S_i \cup \{(C, \alpha)\}$, where (C, α) is the conclusion of the fuzzy linguistic resolution inference

$$\frac{(C_1,\alpha_1),(C_2,\alpha_2)}{(C,\alpha)}$$

 $(C_1, \alpha_1), (C_2, \alpha_2) \in S_i$ and there are not any clauses with reliability $(C_1, \alpha'_1), (C_2, \alpha'_2) \in S_i$ such that $\alpha'_1 > \alpha_1$ and $\alpha'_2 > \alpha_2$, or

- S_{i+1} is obtained by removing a variant with smaller reliability, that is $S_{i+1} = S_i \setminus \{(C, \alpha)\}$ where $(C, \alpha) \in S_i$ and there is some $(C, \alpha') \in S_i$ such that $\alpha < \alpha'$.

Define the limit of a derivation S_0, \ldots, S_i, \ldots

$$S_{\infty} = \bigcup_{i \ge 0} \bigcap_{j \ge i} S_j$$

The following result establishes the soundness and completeness of the resolution procedure.

Theorem 2. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation. S_n contains the empty clause iff S_0 is unsatisfiable (for some $n = 0, 1, \ldots$).

Lemma 2. Consider the following resolution inferences:

$$\frac{(A^a \vee B^{b_1}, \alpha), (B^{b_2} \vee C^c, \beta)}{(A^a \vee C^c, \gamma)}$$
$$\frac{(A^a \vee B^{b_1}, \alpha), (B^{b_2} \vee C^c, \beta')}{(A^a \vee C^c, \gamma')}$$

Then, $\beta' > \beta$ implies $\gamma' \ge \gamma$.

Lemma 3. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation, and S_{∞} be the limit of the derivation. Then S_{∞} is saturated.

Theorem 3. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation, and S_{∞} be the limit of the derivation. Then for each clause (C, α) in S_{∞} , there is not any other resolution proof of the clause (C, α') from S_0 such that $\alpha' > \alpha$.

Example 42 Consider again Example 41. Applying the α -strategy we get the following saturated set of clauses

- 1. $(A^{\mathsf{MFalse}} \lor B^{\mathsf{False}} \lor C^{\mathsf{VMTrue}}, \top)$
- 2. $(B^{\mathsf{LTrue}} \lor C^{\mathsf{PTrue}}, \top)$
- 3. $(A^{\mathsf{PTrue}}, \top)$
- 4. $(B^{\mathsf{VTrue}}, \top)$
- 5. $(C^{\mathsf{VFalse}}, \top)$
- 6. $(B^{\text{False}} \vee C^{\text{VMTrue}}, \text{PTrue})$
- 7. $(A^{\mathsf{MFalse}} \lor C^{\mathsf{VMTrue}}, \mathsf{True})$
- 8. $(A^{\mathsf{MFalse}} \vee C^{\mathsf{VMTrue}}, \mathsf{LTrue})$
- 9. $(A^{\text{MFalse}} \vee B^{\text{False}}, \text{VTrue})$
- 10. $(C^{VMTrue}, PTrue)$
- 11. $(A^{\mathsf{MFalse}}, \mathsf{True})$
- 12. (\Box, PTrue)

The initial set of clauses is unsatisfiable, and the resolution futation is the following

$$\begin{array}{c} (A^{\mathsf{MFalse}} \lor B^{\mathsf{False}} \lor C^{\mathsf{VMTrue}}, \top)(C^{\mathsf{VFalse}}, \top) \\ \hline \\ (A^{\mathsf{MFalse}} \lor B^{\mathsf{False}}, \mathsf{VTrue}) & (B^{\mathsf{VTrue}}, \top) \\ \hline \\ \hline \\ \hline \\ (A^{\mathsf{MFalse}}, \mathsf{True}) & (A^{\mathsf{PTrue}}, \top) \\ \hline \\ \hline \\ (\Box, \mathsf{PTrue}) \end{array}$$

5 Conclusion

We have presented a linguistic logic system with the basic components: syntax, semantics and inference. The syntax have been defined as usual. To define the semantics, the truth value domain have been taken from linear symmetrical hedge algebra and logical connectives have been defined based on Gödel's t-norm and t-conorm. We have also introduced an inference rule associated with a reliability which guarantees that the reliability of the inferred clause is less than or equal to reliaility of the premise clauses. Moreover, we have given a resolution procedure which ensures that the proof of clauses has the maximal reliability. The soundness and completeness of the resolution procedure are also proved. The proofs of the theorems, proposititions and lemmas are omitted due to lack of space. They can be found in the full paper at http://arxiv.org/submit/769464/view. There are several lines of future works. It would be natural to consider the linguistic first order logic in the same settings as our logic here. It would be worth investivating how to extend our result to other hedge algebra structures and to other automated reasong methods.

References

- Rafee Ebrahim. Fuzzy logic programming. Fuzzy Sets and Systems, 117(2):215– 230, 2001.
- Xingxing He, Jun Liu, Yang Xu, Luis Martínez, and Da Ruan. On -satisfiability and its -lock resolution in a finite lattice-valued propositional logic. *Logic Journal* of the IGPL, 20(3):579–588, 2012.
- Jiajun Lai and Yang Xu. Linguistic truth-valued lattice-valued propositional logic system lp(x) based on linguistic truth-valued lattice implication algebra. Inf. Sci., 180(10):1990–2002, 2010.
- Van Hung Le, Fei Liu, and Dinh Khang Tran. Fuzzy linguistic logic programming and its applications. *TPLP*, 9(3):309–341, 2009.
- 5. Richard C. T. Lee. Fuzzy logic and the resolution principle. In *IJCAI*, pages 560–567, 1971.
- B. Mondal and S. Raha. Approximate reasoning in fuzzy resolution. In Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, pages 1–6, 2012.
- Cat-Ho Nguyen and Van-Nam Huynh. An algebraic approach to linguistic hedges in zadeh's fuzzy logic. *Fuzzy Sets and Systems*, 129(2):229–254, 2002.
- Cat-Ho Nguyen, Dinh-Khang Tran, Van-Nam Huynh, and Hai-Chau Nguyen. Hedge algebras, linguistic-valued logic and their application to fuzzy reasoning. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 7(4):347-361, 1999.
- C.H. Nguyen and W. Wechler. Hedge Algebras: An Algebraic Approach in Struture of Sets of Linguistic Truth Values, pages 281–293. Fuzzy Sets and Syst. 35, 1990.
- Le Anh Phuong and Tran Dinh Khang. A deductive method in linguistic reasoning. In Uncertainty Reasoning and Knowledge Engineering (URKE), 2012 2nd International Conference on, pages 137–140, 2012.
- Le Anh Phuong and Tran Dinh Khang. Linguistic reasoning based on generalized modus ponens with linguistic modifiers and hedge moving rules. In *Fuzzy Theory* and it's Applications (iFUZZY), 2012 International Conference on, pages 82–86, 2012.
- Dana Smutná and Peter Vojtás. Graded many-valued resolution with aggregation. Fuzzy Sets and Systems, 143(1):157–168, 2004.
- Peter Vojtás. Fuzzy logic programming. Fuzzy Sets and Systems, 124(3):361–370, 2001.
- Thomas J. Weigert, Jeffrey J. P. Tsai, and Xuhua Liu. Fuzzy operator logic and fuzzy resolution. J. Autom. Reasoning, 10(1):59–78, 1993.
- Yang Xu, Da Ruan, Etienne E. Kerre, and Jun Liu. alpha-resolution principle based on lattice-valued propositional logic lp(x). *Inf. Sci.*, 130(1-4):195–223, 2000.
- 16. Yang Xu, Da Ruan, Etienne E. Kerre, and Jun Liu. alpha-resolution principle based on first-order lattice-valued logic lf(x). *Inf. Sci.*, 132(1-4):221–239, 2001.
- M. Mukaidono Z. Shen, L. Ding. Fuzzy resolution principle. In Proc. 18th Internat. Symp. on Multiple-valued Logic, pages 210–215, 1989.
- 18. Lotfi A. Zadeh. Fuzzy sets. Information and Control, 8(3):338-353, 1965.
- Xiaomei Zhong, Yang Xu, Jun Liu, and Shuwei Chen. General form of -resolution principle for linguistic truth-valued lattice-valued logic. *Soft Comput.*, 16(10):1767– 1781, 2012.

Appendix

A Proof of Propositition 33

Proposition 33. Let A and B be formulae. Then, $A \models B$ iff $\models (A \rightarrow B)$.

Proof. Assume that $A \models B$, for any interpretation I, then if I(A) < W, $I(\neg A) > W$; otherwise if I(A) > W, we recall that $A \models B$, so I(B) > W. Hence, $I(A \rightarrow B) = \neg I(A) \lor I(B) > W$. In other words, $\models (A \rightarrow B)$. Conversely, by a similar way, we can also show that $\models (A \rightarrow B)$ implies $A \models B$.

B Proof of Lemma 1

Lemma 1. The fuzzy linguistic resolution rule 42 is sound.

Proof. We need to prove that for any interpretation I, if $I((A^a \vee B^{b_1}) \wedge (B^{b_2} \vee C^c)) > W$ then $I(A^a \vee C^c) > W$. We have that

$$I((A^a \lor B^{b_1}) \land (B^{b_2} \lor C^c))$$

= $I((A^a \land B^{b_2}) \lor (A^a \land C^c) \lor (B^{b_1} \land B^{b_2}) \lor (B^{b_1} \land C^c))$
= $I(A^a \land B^{b_2}) \lor I(A^a \land C^c) \lor I(B^{b_1} \land B^{b_2}) \lor I(B^{b_1} \land C^c)$

It is easy to show that:

 $- I(A^a \wedge B^{b_2}) \leq I(A^a) \leq I(A^a \vee C^c),$ $- I(A^a \wedge C^c) \leq I(A^a) \leq I(A^a \vee C^c),$ $- I(B^{b_1} \wedge C^c) \leq I(C^c) \leq I(A^a \vee C^c), \text{ and }$ $- I(B^{b_1} \wedge B^{b_2}) \leq W$

So, if $I(A^a \vee C^c) \leq W$ then we must have that

$$I(A^a \wedge B^{b_2}) \vee I(A^a \wedge C^c) \vee I(B^{b_1} \wedge B^{b_2}) \vee I(B^{b_1} \wedge C^c) \leq \mathsf{W}$$

which contradicts with the initial assumption. This completes the proof of the theorem.

C Proof of Theorem 2

Theorem 2. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation. S_n contains the empty clause iff S_0 is unsatisfiable (for some $n = 0, 1, \ldots$).

Proof. First, we prove if S_n contains the empty clause then S_0 is unstatisfialbe.

If S_n contains the empty clause, then S_0 is false under any interpretation. By Lemma 1, we have S_{n-1} is false under any interpretation, too. Similarly, $S_{n-1}, \ldots, S_1, S_0$ are also false under any interpretation. This completes the proof of the soundness of the resolution procedure. We now prove the completeness of the resolution procedure, that means if S_0 is unstatisfialbe then S_n contains the empty clause. We apply the *semantic* tree method for two-valued logics to our linguistic propositional logic.

Let S be a set containing exactly n atoms A_1, A_2, \ldots, A_n . A semantic tree of S is an n-level complete binary tree, each level corresponds to an atom. The left edge of each node at the level i is assigned with the label $A_i \leq W$, and the right edge of each node at the level i is assigned with the label $A_i > W$ (cf.Fig 1).



Fig. 1. Semantic tree

A set of clauses S is failed at the node t of a semantic tree T iff there exist an interpretation I corresponding to a branch in T which contains t, such that S is false under I. A node t is called a *failure node* of S iff S fails at t and does not fail at any node above t. A node t in a semantic tree T is called an *inference node* iff both successor nodes of t are failure nodes. If there are failure nodes for S on every branch of the corresponding semantic tree T, removing all child nodes of each failure node, we receive a *failure tree* FT.

Assume that we have a failure tree FT. Because FT has finite level, so there exists one (or more) leaf node on FT at the highest level, let say this node is called j. Let i be parent node of j. By definition of failure tree, i cannot be failure node. Therefore, i has another child node, named k (Figure 2). If k is a failure node then i is inference node, the lemma is proved. If k is not a failure node then it has two child nodes: l, m. Clearly l, m are at higher level than j. This contradicts with the assumption that j is at the highest level. Therefore k is a failure node and i is an inference node.

Let FT_1 (respectively FT_2) be a failure tree of the set of clauses S_1 (respectively S_2). We denote $FT_1 \supset FT_2$ iff there exists an inference node i of FT_1 such that removing two successor nodes of i on FT_1 we receive FT_2 .

Because S_0 is unsatisfiable, there is a corresponding failure tree FT and an inference node i on FT with two child nodes j, k. Assuming that the label of edge i - j is $A \leq W$ and the label of edge i - k is A > W. The interpretation corresponding to the branch contains the edge i - j and terminating at j makes S_0



Fig. 2. Inference node on failure tree

satisfiable. So, there is at least one clause in S_0 containing the literal A^{α_1} where $\alpha_1 \leq W$, let say C_1 . Similarly, there exists at least one clause in S_0 containing the literal A^{α_2} where $\alpha_2 > W$, we named it C_2 . Applying the resolution rule 42:

$$\frac{C_1, C_2}{C_3}$$

 C_3 does not contain atom A, so that C_3 is false under all interpretations containing A. Thus, failure tree FT_1 of clause set $S_1 = S_0 \cup C_3$ does not contain node j, k, this means $FT \supset FT_1$.

By applying resolution procedure, there exist failure tree FT_2, FT_3, \ldots of the sets of clauses S_2, S_3, \ldots such that $FT \supset FT_1 \supset FT_2 \supset FT_3 \supset \ldots$. Because there are only a finite number of nodes in FT, then exists some n satisfying: $FT_n = \Box$ (i.e. $FT \supset FT_1 \supset FT_2 \supset FT_3 \supset \ldots \supset FT_n = \Box$). Only the empty clause is false under the empty interpretation. This means that the set of clauses S_n (S_n corresponds to FT_n) contains the empty clause. This completes the proof of the theorem.

D Proof of Lemma 2

Lemma 2. Consider the following resolution inferences:

$$\frac{(A^a \vee B^{b_1}, \alpha), (B^{b_2} \vee C^c, \beta)}{(A^a \vee C^c, \gamma)}$$
$$\frac{(A^a \vee B^{b_1}, \alpha), (B^{b_2} \vee C^c, \beta')}{(A^a \vee C^c, \gamma')}$$

Then, $\beta' > \beta$ implies $\gamma' \ge \gamma$.

Proof. We have

 $\gamma = \alpha \land \beta \land \neg (b_1 \land b_2) \land (b_1 \lor b_2)$

 $\begin{aligned} \gamma' &= \alpha \wedge \beta' \wedge \neg (b_1 \wedge b_2) \wedge (b_1 \vee b_2) \\ \text{Denote } \delta &= \min(\alpha, \neg (b_1 \wedge b_2), (b_1 \vee b_2)) \text{ then } \gamma = \min(\beta, \delta), \ \gamma = \min(\beta', \delta). \\ \text{Hence, if } \beta' &> \beta \text{ implies } \gamma' \geq \gamma \\ \text{The equality occurs when } \beta \geq \min(\alpha, \neg (b_1 \wedge b_2), (b_1 \vee b_2)). \end{aligned}$

E Proof of Lemma 3

Lemma 3. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation, and S_{∞} be the limit of the derivation. Then S_{∞} is saturated.

Proof. By contradiction assume that S_{∞} is not saturated. Then there must be a fuzzy linguistic resolution inference

$$\frac{(C_1,\alpha_1),(C_2,\alpha_2)}{(C,\alpha)}$$

where $(C_1, \alpha_1), (C_2, \alpha_2) \in S_{\infty}$, there is not any clause $(C, \alpha') \in S_{\infty}$ such that $\alpha \leq \alpha'$. By definition of α -strategy derivation, either (C, α) is in S_{∞} or there must be a clause (C, α'') in S_i for some $i = 0, 1, \ldots$ such that $\alpha \leq \alpha''$, this also means that (C, α) is removed from S_j for some $j \geq i$. In both cases, we have a contradiction.

F Proof of Theorem 3

Theorem 3. Let S_0, \ldots, S_i, \ldots be a fuzzy linguistic resolution α -strategy derivation, and S_{∞} be the limit of the derivation. Then for each clause (C, α) in S_{∞} , there is not any other resolution proof of the clause (C, α') from S_0 such that $\alpha' > \alpha$.

Proof. By contradiction, suppose that for some clause (C, α) in S_{∞} , there exists a resolution proof of (C, α') from S_0 such that $\alpha' > \alpha$. Let (C_1, α_1) and (C_2, α_2) be the two parents of (C, α') in such a resolution proof of (C, α') . We have that (C_1, α_1) and (C_2, α_2) cannot be both in S_{∞} because otherwise an inference with these two clauses as premisses would give (C, α') in S_{∞} . Without lost of generality, we can assume that (C_1, α_1) is not in S_{∞} . Obviously, the resolution proof of (C_1, α_1) from S_0 can not be in S_{∞} . That also means there is a clause (C_1, α'_1) in S_{∞} such that there is not any other resolution proof of the clause (C_1, α''_1) , where $\alpha''_1 > \alpha'_1$. By Lemma 3, S_{∞} is satutared. According to Lemma 2, the inference with premisses (C_1, α'_1) and (C_2, α_2) gives us the conclusion (C, α'') , with $\alpha'' > \alpha'$. This contradicts with the fact that S_{∞} is satutared. This completes the proof of the theorem.