Applied Mathematics & Information Sciences
An International Journal

A Resolution Method for Linguistic Many-valued Logic

Le Anh Phuong[1] *and Tran Dinh Khang*[2]

[1]Department of Computer Science, Hue University of Education, Hue City, Vietnam
[2]School of Information & Communication Technology, Hanoi University of Science and Technology, Hanoi City, Vietnam

Received: 25 Sep. 2012, Revised: 18 Dec. 2012, Accepted: 20 Jan. 2013
Published online: 1 May 2013

Abstract: This paper studies the linguistic truth value domain (AX) based on finite monotonous hedge algebra and then we extend lukasiewicz algebra on $[0,1]$ to linguistic lukasiewicz algebra on linguistic truth value domain (AX), in an attempt to propose a general resolution for linguistic many-valued logic based on hedge moving rules and linguistic lukasiewicz algebra for linguistic reasoning. Its theorems of soundness and completeness associated with general resolution are also proved. This reflects the symbolic approach acts by direct reasoning on linguistic truth value domain.

Keywords: Hedge algebra, linguistic truth value domain, general resolution, linguistic many-valued logic.

1. Introduction

Automated reasoning based on Robinson's resolution rule have been extensively studied in the context of finding natural and efficient proof systems to support a wide spectrum of computational tasks. With the development of Zadeh's fuzzy logic [1, 2], expert systems and knowledge engineering, especially since non-classical logics have become a considerable formal tool for computer science and artificial intelligence. The area of automated reasoning based on non-clasical logic has drawn the attention of many research, especially for fuzzy logic and for multiple-valued logic.

Studied in [11–15] are proposed a resolution method for fuzzy logic, multiple-valued logic based on truth values on $[0,1]$ and lattice respectively to solve the problem of approximate reasoning.

Based on hedge algebraic (HA) structures in order to model the linguistic truth value domain was studied in [3, 4], it has been analyzed and proposed in [9] that monotonous hedge algebra be used for the processing systems using hedge moving rules in combination with fuzzy reasoning which satisfy semantic inheritance and accommodation. Based on monotonous algebra, one can build inverse mapping of hedges with limited length [8], allowing the expansion of hedge moving rules.

Based on the results in [9], the writing suggests finite monotonous hedge algebra be the linguistic truth value domain for linguistic many-valued logic, using hedge moving rules, hedge inverse mapping and general resolution based on linguistic lukasiewicz algebra for linguistic many-valued logic to solve the problem of reasoning.

The paper consists of five parts: the preliminaries followed by Section 2, presenting basic knowledge serving as theoretical foundation for the research. Section 3 is for research in linguistic many-valued logic based on the linguistic truth value domain. Section 4 describes the resolution method for linguistic many-valued logic based on linguistic lukasiewicz algebra and hedge moving rules for linguistic reasoning. The last section is the conclusion.

2. Preliminaries

In this session, we would present some concepts, properties of the monotonous hedge algebra, hedge inverse mapping that have been researched in [3–5, 8–10].

2.1. Monotonous hedge algebra

Consider a truth domain consisting of linguistic values, e.g., *VeryVeryTrue*, *PossiblyMoreFalse*; etc. In such a truth domain the value *VeryVeryTrue* is obtained by applying the modifier *Very* twice to the generator *True*. Thus, given a set of generators $G = (True; False)$ and a

* Corresponding author e-mail: leanhphuong@dhsphue.edu.vn

nonempty finite set H of hedges, the set X of linguistic values is $\{\delta c | c \in G, \delta \in H^*\}$.

Furthermore, if we consider $True > False$, then this order relation also holds for other pairs, e.g., $VeryTrue > MoreTrue$. It means that there exists a partial order $>$ on X.

In general, given nonempty finite sets G and H of generators and hedges resp., the set of values generated from G and H is defined as $X = \{\delta c | c \in G, \delta \in H^*\}$. Given a strictly partial order $>$ on X, we define $u \geq v$ if $u > v$ or $u = v$. Thus, X is described by an abstract algebra HA = $(X, G, H, >)$.

Each hedge $h \in H$ can be regarded as a unary function $h : X \to X; x \mapsto hx$. Moreover, suppose that each hedge is an ordering operation, i.e., $\forall h \in H, \forall x \in X : hx > x$ or $hx < x$. Let $I \notin H$ be the identity hedge, i.e., $Ix = x$ for all $x \in X$. Let us define some properties of hedges in the following definition.

Definition 1. *A hedge chain σ is a word over H, $\sigma \in H^*$. In the hedge chain $h_p \dots h_1$, h_1 is called the first hedge whereas h_p is called the last one. Given two hedges $h; k$, we say that:*

i) h and k are converse if $\forall x \in X$: $hx > x$ iff $kx < x$;
ii) h and k are compatible if $\forall x \in X$: $hx > x$ iff $kx > x$;
iii) h modifies terms stronger or equal than k, denoted by $h \geq k$, if $\forall x \in X$: $(hx \leq kx \leq x)$ or $(hx \geq kx \geq x)$; $h > k$ if $h \geq k$ and $h \neq k$;
iv) h is positive w.r.t. k if $\forall x \in X$: $(hkx < kx < x)$ or $(hkx > kx > x)$;
v) h is negative w.r.t. k if $\forall x \in X$: $(kx < hkx < x)$ or $(kx > hkx > x)$;

The most commonly used HA are symmetric ones, in which there are exactly two generators, like e.g., $G = \{True; False\}$. In this paper, we only consider symmetric HA. Let $G = \{c^+, c^-\}$, where $c^+ > c^-$. c^+ and c^- are called positive and negative generators respectively. The set H is decomposed into the subsets $H^+ = \{h \in H | \ hc^+ > c^+\}$ and $H^- = \{h \in H | hc^+ < c^+\}$. For each value $x \in X$, let $H(x) = \{\sigma x | \sigma \in H^*\}$.

Definition 2. *An abstract algebra $(X, G, H, >)$, where $H \neq \emptyset$, $G = \{c^+, c^-\}$ and $X = \{\sigma c | c \in G, \sigma \in H^*\}$, is called a linear symmetric HA if it satisfies the following conditions:*

A1) For all $h \in H^+$ and $k \in H^-$, h and k are converse.
A2) The sets $H^+ \cup \{I\}$ and $H^- \cup \{I\}$ are linearly ordered with the least element I.
A3) For each pair h, $k \in H$, either h is positive or negative wrt k.
A4) If $h \neq k$ and $hx < kx$ then $h'hx < k'kx$, for all h, k, h', $k' \in H$ and $x \in X$.
A5) If $u \notin H(v)$ and $u < v$ ($u > v$) then $u < hv$ ($u > hv$, resp.), for any $h \in H$.

Example 1. Consider a HA $(X, \{True; False\}, H, >)$, where $H = \{Very, More, Probably, Mol \ (More$ or $Less)\}$, and (i) $Very$ and $More$ are positive wrt $Very$ and $More$, negative wrt $Probably$ and Mol; (ii) $Probably$ and Mol are negative wrt $Very$ and $More$, positive wrt $Probably$ and Mol.

H is decomposed into $H^+ = \{Very, More\}$ and $H^- = \{Probably, Mol\}$. In $H^+ \cup \{I\}$ we have $Very > More > I$, whereas in $H^- \cup \{I\}$ we have $Mol > Probably > I$.

Definition 3. ***(Mono-HA)*** *A HA $(X; G; H; >)$ is called monotonic if each $h \in H^+$ (H^-) is positive wrt all $k \in H^+$ (H^-), and negative wrt all $h \in H^-$ (H^+).*

As defined, both sets $H^+ \cup \{I\}$ and $H^- \cup \{I\}$ are linearly ordered. However, $H \cup \{I\}$ is not, e.g., in Example 1 $Very \in H^+$ and $Mol \in H^-$ are not comparable. Let us extend the order relation on $H^+ \cup \{I\}$ and $H^- \cup \{I\}$ to one on $H \cup \{I\}$ as follows.

Definition 4. *Given $h, k \in H \cup \{I\}$, $h \geq_h k$ iff*

i) $h \in H^+$, $k \in H^-$; or
ii) $h, k \in H^+ \cup \{I\}$ and $h \geq k$; or
iii) $h, k \in H^- \cup \{I\}$ and $h \leq k$.
($h >_h k$ iff $h \geq_h k$ and $h \neq k$)

Example 2. The HA in Example 1 is Mono-HA. The order relation $>_h$ in $H \cup \{I\}$ is $Very >_h More >_h I >_h Probably >_h Mol$.

Then, in Mono-HA, hedges are "context-free", i.e., a hedge modifies the meaning of a linguistic value independently of preceding hedges in the hedge chain.

2.2. Inverse mapping of hedge

In application of hedge algebra into direct reasoning on natural language [4], using hedge moving rule $RT1$ and $RT2$:

$$RT1 : \frac{(p\,(x; hu)\,, \delta c)}{(p\,(x; u)\,, \delta hc)}; \qquad RT2 : \frac{(p\,(x; u)\,, \delta hc)}{(p\,(x; hu)\,, \delta c)}$$

Example 3. Applying rule of hedge moving, there are two equal statements: "It is true that Robert is very old" and "It is very true that Robert is old". It means that if the reliability of the sentence: "Robert is very old" is "True", the reliability of the sentence: "Robert is old" is "Very True" and vice versa.

However the above hedge moving rules are not applied in such case as from the true value of the sentence: "John is young" is "Very True" , we can not count the true value of the sentence: "John is more young". To overcome the above weak point, in [5–7] inverse mapping of hedge is proposed.

Definition 5. *Given Mono-HA $= (X, \{c^+, c^-\}, H, \leq)$ and hedge $h \in H$. We take $AX = X \cup \{0, W, 1\}$ of which 0, W, 1 are the smallest, neutral, and biggest element in AX respectively. A mapping h^-: $AX \to AX$ is called inverse mapping of h if it meets the following conditions:*

i) $h^-(\delta hc) = \delta c$ *of which* $c \in G = \{c^+, c^-\}$, $\delta \in H^*$
ii) $x \le y \Rightarrow h^-(x) \le h^-(y)$ *of which* $x, y \in X$

In case of inverse mapping of a hedge string, we determine it, based on inverse mapping of single hedges as follows:

$$(h_k, h_{k-1}, \ldots, h_1)^- (\delta c) = h_k^- \left(\ldots \left(h_1^- (\delta c) \right) \ldots \right)$$

Then the rule (RT2) is generalized as follows:

$$GRT2 : \frac{(p(x;u), \delta c)}{(p(x;hu), h^-(\delta c))}$$

In [5–8], it is shown that inverse mapping of hedge always exists and inverse mapping value of hedge is not unique.

3. Linguistic many-valued logic

3.1. Linguistic truth value domain

In real life, people only use a string of hedge with a finite length for a vague concept in order to have new vague concepts and only use a finite string of hedges for truth values. This makes us think about limiting the length of the hedge string in the truth value domain to make it not exceed L-any positive number. In case that intellectual base has a value having length of hedge string bigger than L, we need to approximate the value having hedge string $\le L$. Based on monotonous hedge algebra Mono-HA, we set finite monotonous hedge algebra to make linguistic truth value domain.

Definition 6. ***(L-Mono-HA)*** *L-Mono-HA, L is a natural number, is a Mono-HA with standard presentation of all elements having the length not exceeding* $L + 1$.

Definition 7. ***(Linguistic truth value domain)*** *A linguistic truth value domain* AX *taken from a L-Mono-HA* $= (X, \{c^+, c^-\}, H, \le)$ *is defined as* $AX = X \cup \{0, W, 1\}$ *of which* 0, W, 1 *are the smallest, neutral, and biggest elements respectively in* AX.

Example 4. Given finite monotonous hedge algebra 2-Mono -HA $= (X, \{c^+, c^-\}, \{V, M, P\}, \le)$ ($V = Very$; $M = More$; $P = Possibly$) ($P \in H^-$, $M, V \in H^+$, $M < V$).

We have the linguistic truth value domain:
$AX = \{0, VVc^-, MVc^-, Vc^-, PVc^-, VMc^-, MMc^-, Mc^-, PMc^-, c^-, VPc^-, MPc^-, Pc^-, PPc^-, W, PPc^+, c^+, MPc^+, VPc^+, c^+, PMc^+, Mc^+, MMc^+, VMc^+, PVc^+, Vc^+, MVc^+, VVc^+, 1\}$.

Proposition 1 *If we have L-Mono-HA* $= (X, \{c^+, c^-\}, H, \le)$*, the linguistic truth value domain* AX *is finite to a number of elements* $|AX| = 3 + 2\sum_{i=0}^{L} |H|^i$ *and elements of* AX *is linearly ordered. (The symbol* $|AX|$ *is the number of elements of* AX *and* $|H|$ *is the number of* H*).*

Proof. Suppose that $|H| = n$, we always have 3 elements 0, 1, W;

With $i = 0$, we have 2 more elements $\{c^+, c^-\}$; $i = 1$, we have $2n^1$ more elements; with $i = L$ we have $2n^L$ more elements.

Then
$|AX| = 3 + 2\left(1 + n + \ldots + n^L\right) = 3 + 2\sum_{i=0}^{L} |H|^i$

According to the definition of linear order relation in monotonous hedge algebra Mono-HA, we see that, elements in AX are linearly ordered.

Example 5. According to Example 4, we have the language true value domain (is linearly ordered) $AX = \{v_1 = 0, v_2 = VVc^-, v_3 = MVc^-, v_4 = Vc^-, v_5 = PVc^-, v_6 = VMc^-, v_7 = MMc^-, v_8 = Mc^-, v_9 = PMc^-, v_{10} = c^-, v_{11} = VPc^-, v_{12} = MPc^-, v_{13} = Pc^-, v_{14} = PPc^-, v_{15} = W, v_{16} = PPc^+, v_{17} = Pc^+, v_{18} = MPc^+, v_{19} = VPc^+, v_{20} = c^+, v_{21} = PMc^+, v_{22} = Mc^+, v_{23} = MMc^+, v_{24} = VMc^+, v_{25} = PVc^+, v_{26} = Vc^+, v_{27} = MVc^+, v_{28} = VVc^+, v_{29} = 1\}$.

We can determine the index of v by Algorithm 1:

Algorithm 1 Finding index

```
Input: Domain (Truth) of $L - mono - HA$ is $AX$,
$H^- = \{h_{-q}, \ldots, h_{-1}\}$, $H^+ = \{h_1, \ldots h_p\}$
$x = l_k l \ldots l_1 c$ with $c \in \{T, F\}$, $k \le L$
Output: Finding $index$ of $v$ so that $v_{index} = x$
Methods: $M = 3 + 2\sum_{i=0}^{L} (p+q)^i$;
if $x = 0$ then $index = 1$;
if $x = W$ then $index = (M+1)/2$;
if $x = 1$ then $index = M$;
$index = (M+1)/2 + 1 + qAX^1$;
for $i = 1$ to $k - 1$ do
  {find $j$ such that $l_i = h_j$
  if $j > 0$ then
    $index = index + (j-1)\left|AX^i\right| + q\left|AX^{i+1}\right| + 1$;
  if $j < 0$ then
    $index = index - (|j|-1)\left|AX^i\right| - p\left|AX^{i+1}\right| - 1$;
  }
find $j$ such that $l_k = h_j$ /*$j > 0$ then $l_k \in H^+$, else $l_k \in H^-$*/
if $k < L$ then
  {if $j > 0$ then
    $index = index + (j-1)\left|AX^k\right| + q\left|AX^{k+1}\right| + 1$;
  if $j < 0$ then
    $index = index + (|j|-1)\left|AX^k\right| + q\left|AX^{k+1}\right| - 1$;
  }
else $index = index + j$;
if $c = False$ then $index = (M+1) - index$;
return ($index$)
/*$|AX| = \sum_{k=0}^{L-i} (p+q)^k$*/
```

Based on the algorithm to identify the inverse map of hedge and properties studied in [8], we can establish the inverse map for 2-Mono-HA $= (X, \{c^+, c^-\}, \{V, M, P\}, \le)$ with a note that, if $h^-(x) = W$ with $x \in H(c^+)$ we can consider

Table 1 Inverse mapping of hedges

	V^-	M^-	P^-
0	0	0	0
kVc^-	VVc^-	VVc^-	kVc^-
kMc^-	VVc^-	kVc^-	Mc^-
c^-	Vc^-	Mc^-	c^-
VPc^-	VMc^-	PMc^-	MPc^-
MPc^-	MMc^-	Pc^-	MPc^-
Pc^-	Mc^-	Pc^-	Pc^-
PPc^-	PMc^-	VPc^-	Pc^-
W	W	W	W
PPc^+	PPc^+	PPc^+	VPc^+
Pc^+	$Pc^\mp$	MPc^+	c^+
MPc^+	MPc^+	VPc^+	Mc^+
VPc^+	MPc^+	VPc^+	MMc^+
c^+	VPc^+	c^+	VMc^+
kMc^+	c^+	kMc^+	kVc^+
kVc^+	kMc^+	VMc^+	VVc^+
1	1	1	1

$h^-(x) = VPc^+$ the smallest value of $H(c^+)$; if $h^-(x) = 1$ with $x \in H(c^+)$ we can consider $h^-(x) = VVc^+$ the biggest value of $H(c^+)$; If $h^-(x) = W$ with $x \in H(c^-)$ we can consider $h^-(x) = VPc^-$ the biggest value of $H(c^-)$; if $h^-(x) = 0$ with $x \in H(c^-)$ we can consider $h^-(x) = VVc^-$ the smallest value of $H(c^-)$.

The following is an example on inverse map of 2-Mono-HA= $(X, \{c^+, c^-\}, V, M, P, \leq)$, $(k \in H)$ (see Table 1).

3.2. Linguistic lukasiewicz algebra

In logic, the truth value domain is shown by a algebra structure with calculations $\wedge, \vee, \neg, \rightarrow$. Many-valued logic has finite truth value domain including elements according to linearly order on $[0, 1]$ and lukasiewicz algebra is an algebra structure for this truth value domain.

Definition 8. *[10] **(Lukasiewicz algebra)** The structure $\mathcal{L} = ([0, 1], \wedge, \vee, \otimes, \oplus, \neg, \rightarrow, 0, 1)$ is called as lukasiewicz algebra with $[0, 1]$ which is segment of real numbers between 0 and 1, 0 is the smallest value element, 1 is the biggest value element and $\wedge$, $\vee$, $\otimes$, $\oplus$, $\neg$, $\rightarrow$ are operators defined as follows: (with $a, b \in [0, 1]$)*

i) $a \wedge b = \min(a, b)$

ii) $a \vee b = \max(a, b)$

iii) $a \rightarrow b = \min(1, 1 - a + b)$

iv) $\neg a = 1 - a$

v) $a \otimes b = \max(0, a + b - 1)$

vi) $a \oplus b = \min(1, a + b)$

We have the linguistic truth value domain $AX = \{v_i, i = 1, 2, \ldots, n\}$ with $v_1 = 0$ and $v_n = 1$ in finite monotonous hedge algebra and linear order or $AX = \{v_i, i = 1, 2, \ldots, n;\ v_1 = 0, v_n = 1$ and $\forall 1 \leq i, j \leq n : v_i \geq v_j \Leftrightarrow i \geq j\}$

Based on definition above, we can extend $[0, 1]$ to AX, when we have the definition following:

Definition 9. ***(Linguistic lukasiewicz algebra)*** *The structure $\mathcal{L}_n = (AX, \wedge, \vee, \otimes, \oplus, \neg, \rightarrow, 0, 1)$ is called as linguistic lukasiewicz algebra with AX which is segment of real numbers between 0 and 1, 0 is the smallest value element, 1 is the biggest value element and $\wedge$, $\vee$, $\otimes$, $\oplus$, $\neg$, $\rightarrow$ are operators defined as follows: (with $v_i, v_j \in AX$)*

i) $v_i \vee v_j = v_{\max\{i,j\}}$

ii) $v_i \wedge v_j = v_{\min\{i,j\}}$

iii) $\neg v_i = v_{n-i+1}$

iv) $v_i \rightarrow_{\mathcal{L}} v_j = v_{\min\{n, n-i+j\}}$

v) $v_i \otimes v_j = v_1 \vee v_{i+j-n}$

vi) $v_i \oplus v_j = v_n \wedge v_{i+j}$

The biresiduation operation $\leftrightarrow$ could be defined $v_i \leftrightarrow v_j =_{df} (v_i \rightarrow v_i) \wedge (v_j \rightarrow v_i)$. The following properties of $\mathcal{L}_n$ will be used in the sequel:

$$v \otimes 1 = v;\ \ v \otimes 0 = 0;\ \ v \oplus 1 = 1;\ \ v \oplus 0 = v$$
$$v \rightarrow 1 = 1;\ \ v \rightarrow 0 = \neg v;\ \ 1 \rightarrow 0 = v;\ \ 0 \rightarrow v = 1$$

Lemma 1 *Let $v_a, v_b, v_c \in AX$, we have:*

$$(v_a \rightarrow v_b) \otimes (v_b \rightarrow v_c) \leq v_a \rightarrow v_c$$

Proof. We have:

$$v_a \rightarrow v_b = v_{\min\{n, n-a+b\}}$$
$$v_b \rightarrow v_c = v_{\min\{n, n-b+c\}}$$
$$v_a \rightarrow v_c = v_{\min\{n, n-a+c\}}$$

Otherwise:

$$\min(n, n - a + b) = \frac{|2n - a + b|}{2} - \frac{|a - b|}{2},$$
$$\min(n, n - b + c) = \frac{|2n - b + c|}{2} - \frac{|b - c|}{2}$$

Because of:

$$\min(n, n - a + b) + \min(n, n - b + c) - n$$
$$= \frac{|2n - a + b|}{2} - \frac{|a - b|}{2} + \frac{|2n - b + c|}{2} - \frac{|b - c|}{2} - n$$
$$= \frac{|2n - a + c|}{2} - \left(\frac{|a - b|}{2} + \frac{|b - c|}{2}\right)$$
$$\leq \frac{|2n - a + c|}{2} - \frac{|a - c|}{2} = \min(n, n - a + c)$$

So, we have:

$$(v_a \rightarrow v_b) \otimes (v_b \rightarrow v_c)$$
$$= v_{\min\{n,n-a+b\}} + v_{\min\{n,n-b+c\}} - n \leq v_a \rightarrow v_c$$

3.3. Linguistic many-valued logic

Many-valued logic is a generalization of Boolean logic. It provides truth values that are intermediate between *True* and *False*. We denote by N the number of truth degrees in many-valued logic.

The linguistic truth value domain $AX = \{v_i, i = 1, 2, \ldots, n\}$ with $v_1 = 0$ and $v_n = 1$ in finite monotonous hedge algebra and linear order or $AX = \{v_i, i = 1, 2, \ldots, n; v_1 = 0, v_n = 1$ and $\forall 1 \leq i, j \leq n$: $v_i \geq v_j \Leftrightarrow i \geq j\}$.

In linguistic many-valued logic, an assertion is one pair $A = (p(x; u), \delta c)$ (Symbol: (A, v)), herein x is a variable, u is a vague concept, δ is the hedge strings, $p(x; u)$ is a vague sentence, δc is a linguistic truth value and $\delta c \in AX$. In this context, the following equivalence holds:

$$(p(x; hu), \delta c) \Leftrightarrow (p(x; u), \delta hc) \qquad \text{(RT1)}$$
$$(p(x; u), \delta c) \Leftrightarrow (p(x; hu), h^{-}(\delta c)) \quad \text{(GRT2)}$$

(With h is a hedge and $\delta hc, h^{-}(\delta c) \in AX$)

Based on linguistic lukasiewicz algebra, the syntax and semantic of linguistic many-valued logic is following:

Syntax: propositional variables, logical constants $-\{a|\ a \in AX\}$. Instead of 0 we write $\perp$ and instead of 1 we write $\top$, connectives & (Lukasiewicz conjunction), $\wedge$ (conjunction), ∇ (Lukasiewicz disjunction), $\vee$ (disjunction), $\Rightarrow$ (implication), $\Leftrightarrow$ (equivalence), $\neg$ (negation) and futuremore by F_J we denote set of all formulas of linguistic many-valued logic in language J.

Semantic: connectives have the following semantic interpretations:
$D(a) = a$ for $a \in AX$,
$D(A\&B) = D(A) \otimes D(B)$,
$D(A \wedge B) = D(A) \wedge D(B)$,
$D(A \nabla B) = D(A) \oplus D(B)$,
$D(A \vee B) = D(A) \vee D(B)$,
$D(A \Rightarrow B) = D(A) \rightarrow D(B)$,
$D(A \Leftrightarrow B) = D(A) \leftrightarrow D(B)$,
$D(\neg A) = \neg D(A)$.

4. Resolution method for linguistic many-valued logic

4.1. Knowledge base systems

In linguistic many-valued logic, an assertion is one pair $A = (p(x; u), \delta c)$, herein x is a variable, u is a vague concept, δ is the hedge strings, $p(x; u)$ is a vague sentence, δc is a linguistic truth value and $\delta c \in AX$. It will be written as (A, v), where A is formula and v is syntactic evaluation. In this context, the following equivalence holds:

$$(p(x; hu), \delta c) \Leftrightarrow (p(x; u), \delta hc) \qquad \text{(RT1)}$$
$$(p(x; u), \delta c) \Leftrightarrow (p(x; hu), h^{-}(\delta c)) \quad \text{(GRT2)}$$

(With h is a hedge and $\delta hc, h^{-}(\delta c) \in AX$)

One knowledge base K in linguistic many-valued logic is a finite set of assertions (A, v). From the given knowledge base K, we can deduce new assertions by using the resolution method.

4.2. Resolution method

In the previous research [13] referred to general fuzzy resolutions with truth values on interval $[0, 1]$, In the following tasks, We extend the results in [13] for linguistic many-valued logic to solve linguistic reasoning.

Definition 10. ***(Inference rule)*** *An n-ary inference rule r in graded logical system is a scheme*

$$r: \frac{(A_1, v_1), \ldots, (A_n, v_n)}{\left(r^{syn}\left(A_1, \ldots, A_n\right), r^{evl}\left(v_1, \ldots, v_n\right)\right)}$$

Using which the evaluated formulas $(A_1, v_1), \ldots, (A_n, v_n)$ are assigned the evaluated formula $\left(r^{syn}\left(A_1, \ldots, A_n\right),\ r^{evl}\left(v_1, \ldots, v_n\right)\right)$. The syntactic operation r^{syn} is a partial n-ary operation on F_J and the evaluation operation r^{evl} is an n-ary lower semicontinous operation on L (i.e. it preserves arbitrary suprema in all variables).

Definition 11. ***(Formal theory for linguistic many-valued logic)*** *A formal theory T in language J is a triple:*

$$T = \langle LAx, SAx, R\rangle$$

Where $LAx \subset F_J$ is a set of logical axioms, $SAx \subset F_J$ is a set of special axioms, and R is a set of sound inference rules.

Definition 12. ***(Evaluated formal proof)*** *An evaluated formal proof of formula A from the set $X \subset F_J$ is a finite sequence of evaluated formulas*

$$w := (A_1, v_1), \ldots, (A_n, v_n) \quad (*)$$

Such that $A_n := A$ and for each $i \leq n$, either there exists an n-ary inference rule r such that

$$(A_i, v_i) := \left(r^{syn}\left(A_{r_1}, \ldots, A_{r_n}\right), r^{evl}\left(v_{r_1}, \ldots, v_{r_n}\right)\right), \quad i_1, \ldots, i_n < n$$

or

$$(A_i, v_i) := (A_i, X(A_i))$$

We will denote the value of evaluated proof by $Val(w) = v_j$, which is the value of the last member in (*).

Definition 13. ***(Provability and truthfulness)*** *Let T be a formal theory and $A \in F_J$ a formula. We write $T \vdash_a A$ and say that the formula A is theorem in the degree a, or provable in the degree a in the formal theory T.*

$T \vdash_a A$ iff $a = \vee\{Val(w)|\ w$ is a proof of A from $LAx \cup SAx\}$

We write $T \vDash_a A$ and say that the formula A is true in the degree a in the formal theory T.

$T \vDash_a A$ iff $a = \wedge\{D(A)|D \vDash T\}$, where the condition $D \vDash T$ hold, if for every $A \in LAx$: $LAx(A) \leq D(A)$, $A \in SAx$: $SAx(A) \leq D(A)$.

The extending fuzzy modus ponens rule for linguistic many-valued logic could be formulated:

Definition 14. ***(Modus ponens for linguistic many-valued logic)***

$$r_{MP} : \frac{(A, a), (A \Rightarrow B, b)}{(B, a \otimes b)}$$

Where from premise A holding in the degree a and premise $A \Rightarrow B$ holding in the degree b we infer B holding the degree $a \otimes b$. r_{MP} could be viewed as a special case of resolution in classical logic. From this fact, the completeness of system based on resolution can be deduced. It is possible to introduce following notion of resolution, with respect to the modus ponens:

Definition 15. ***(General resolution for linguistic many-valued logic)***

$$r_R : \frac{(F_1[G], a), (F_2[G], b)}{(F_1[G/\bot] \nabla F_2[G/\top], a \otimes b)}$$

Where F_1 holding in the degree a and F_2 holding in the degree b are formulas premises of linguistic many-valued logic and G is an occurrence as a subformula in $F_1$and F_2. The expression below the line means the resolvent of premises on G holding in the degree $a \otimes b$. Every occurrence of G is replaced by false in the first formula and by true in the second one.

Theorem 1. ***(Soundness of r_R)*** *The inference r_R rule for linguistic many-valued logic based on $\mathcal{L}_n$ is sound i.e. for every truth valuation D*

$$D\left(r^{syn}(A_1, \ldots, A_n)\right) \geq r^{evl}\left(D(A_1), \ldots, D(A_n)\right)$$

hold true

Proof. For r_R we may rewrite the values of the left and right parts of equation above:

$$D\left(r^{syn}(A_1, \ldots, A_n)\right) = D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right]$$
$$r^{evl}(D(A_1), \ldots, D(A_n)) = D(F_1[G]) \otimes D(F_2[G])$$

It is sufficient to prove the equality for $\Rightarrow$ since all other connectives could be defined by it. By induction on the complexity of fomula $|A|$, defined as the number of connectives, we can prove:

Let premises F_1 and F_2 be atomic fomular. Since they must contain the same subformula then $F_1 = F_2 = G$ and it holds:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right] = D(\bot \nabla \top) = 0 \oplus 1 = 1 \geq D(F_1[G]) \otimes D(F_2[G])$$

Induction step: Let primises F_1 and F_2 be complex formulas and let A and B are subformulas of F_1, C and D are subformulas of F_2 and G is an atom here generally $F_1 = (A \Rightarrow B)$ and $F_2 = (C \Rightarrow D)$. The complexity of $|F_1| = |A| + 1$ or $|F_1| = |B| + 1$ and $|F_2| = |C| + 1$ or $|F_2| = |D| + 1$. Since they must contain the same subformulas and for A, B, C, D the induction presupposition hold it remain to analyze the following cases:

1.$F_1 = (A \Rightarrow G)$, $F_2 = (G \Rightarrow D)$:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right]$$
$$= D([A \Rightarrow \bot] \nabla [\top \Rightarrow D]) = D(\neg A \nabla D)$$
$$= v_n \wedge v_{n+1-a+d} = v_{\min\{n, n+1-a+d\}}$$

We have rewrite the expression in to lukasiewicz interpretation. Now we will try to rewrite the right side of the inequality, which has to be proven.

$D(F_1[G]) \otimes D(F_2[G]) = D(A \Rightarrow G) \otimes D(G \Rightarrow D)$
$= (D(A) \rightarrow D(G)) \otimes (D(G) \rightarrow D(D))$
$\leq D(A) \rightarrow D(D) = v_{\min\{n, n-a+d\}}$ (According to Lemma 1). So, we have:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right] \geq D(F_1[G]) \otimes D(F_2[G])$$

2.$F_1 = (A \Rightarrow G)$, $F_2 = (C \Rightarrow D)$:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right]$$
$$= D([A \Rightarrow \bot] \nabla [C \Rightarrow \top])$$
$$= 1 \geq D(F_1[G]) \otimes D(F_2[G])$$

3.$F_1 = (G \Rightarrow B)$, $F_2 = (G \Rightarrow D)$:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right]$$
$$= D([\bot \Rightarrow B] \nabla [\top \Rightarrow D])$$
$$= 1 \geq D(F_1[G]) \otimes D(F_2[G])$$

4.$F_1 = (G \Rightarrow B)$, $F_2 = (C \Rightarrow G)$:

$$D\left[D(F_1[G/\bot]) \nabla D(F_2[G/\top])\right]$$
$$= D([\bot \Rightarrow B] \nabla [C \Rightarrow \top])$$
$$= 1 \geq D(F_1[G]) \otimes D(F_2[G])$$

By induction we have proven that the inequality holds and the r_R is sound. The induction of the case where only one of the premises has greater complexity is included in the above solved induction step.

From this result we can conclude the completeness theorem. We will need two additional simplification rules for purposes of proof:

Definition 16. ***(Simplification rules for $\nabla, \Rightarrow$)***

$$r_{s\nabla} : \frac{(\bot \nabla A, a)}{(A, a)} \quad \textit{and} \quad r_{s\Rightarrow} : \frac{(\top \Rightarrow A, a)}{(A, a)}$$

The soundness of $r_{s\nabla}$ and $r_{s\Rightarrow}$ is straightforward.

Theorem 2. ***(Completeness for linguistic many - valued logic with*** r_R***,*** $r_{s\nabla}$***,*** $r_{s\Rightarrow}$ ***instead of*** r_{MP}***)*** *Formal theory, where* r_{MP} *is replaced with* r_R*,* $r_{s\nabla}$*,* $r_{s\Rightarrow}$ *is complete i.e. for every* A *from the set of formulas* $T \vdash_a A$ *iff* $T \models_a A$*.*

Proof. The left to right implication (soundness of such formal theory) could be easily done from the soundness of the resolution rule. Conversely it is sufficient to prove that rule r_{MP} can be replaced by r_R, $r_{s\nabla}$, $r_{s\Rightarrow}$. Indeed, let w be a proof:
$w := (A, a)$ {a proof w_a}, $(A \Rightarrow B, b)$ { a proof $w_{A\Rightarrow B}$}, $(B, a \otimes b)$ $\{r_{MP}\}$

Then we can replace it by proof:
$w := (A, a)$ { a proof w_a}, $(A \Rightarrow B, b)$ { a proof $w_{A\Rightarrow B}$}, $(\bot \nabla [\top \Rightarrow B], a \otimes b)$ $\{r_R\}$, $(\top \Rightarrow B, a \otimes b)$ $\{r_{s\nabla}\}$, $(B, a \otimes b)$ $\{r_{s\Rightarrow}\}$

Using the last sequence we can easily make a proof with r_{MP} also with the proposed r_R and simplification rules. Since usual formal theory with r_{MP} is complete as it is proved in [10], every formal theory with these rules is else complete.

4.3. Deductive procedure

The deduction method is derived from knowledge base K using the above rules to deduce the conclusion (P, v), we can write $K \vdash (P, v)$. Let $C(K)$ denote the set of all possible conclusions: $C(K) = \{(P, v) : K \vdash (P, v)\}$. A knowledge base K is called consistent if, from K, we can not deduce two assertions (P, v) and $(\neg P, v)$.

Here, we build an deduction procedure (Algorithm 2) based on hedge moving rules and r_R for solving the linguistic reasoning problem.

Problem: Suppose that we have a given knowledge base K, how can we deduce conclusions from K?

Algorithm 2 Deductive procedure

Input: Knowledge base set K; L-Mono-HA
Output: Truth value of the conclusion clause.
Methods:

```
    Step 1: Using the moving rules $RT1$ and $GRT2$ to determine the dim unknown claims in the knowledge base. In the case of the linguistic truth value of the new clause does not belong to $AX$, or the hedge series length is greater than $L$, we must approximate the hedge series to hedge series of length $L$ by removing the outside left hedge. (The outside left hedge of hedge series make little change to the semantics of linguistic truth value);
    Step 2: Transferring the truth value $\delta c$ in the expression found in Step 2 into $v_i$ : $v_i = \delta c$ (Algorithm 1);
    Step 3: Calculating the truth value of formula based on $r_R$, $r_{s\nabla}$, $r_{s\Rightarrow}$;
    Step 4: Making the truth value of conclusion clause.
```

Example 6. Given the following knowledge base:

i) If a student studying more hard and his university is high-raking, then he will be a good employee is more very true.
ii) The university where Mary studies is very high-raking is possibly true.
iii) Mary is studying very hard is more true.

Find the truth value of the sentence: "Mary will be a good employee"

By formalizing. (i) - (iii) an be rewritten by follow:
$(studying(x; MHard) \wedge is(Univ(x); Hi\text{-}ra)$
$\to emp(x; good)), MVTrue)$
(Base on the hypothesis (i))
$(is(Univ(Mary); VHi\text{-}ra), PTrue))$
$\equiv$ $(is(Univ(Mary); Hi\text{-}ra), PVTrue))$ (ii and RT1)
(Base on (ii))
$(studying(Mary; VHard), MTrue)$
$\equiv$ $(studying(Mary; MHard), M^{-}(MVTrue))$ (iii, RT1 and GRT2)
(Base on (iii))

Set: $a = (studying(x; MHard)$; $b = is(Univ(x); Hi\text{-}ra)$; $c = emp(x; good)$. Based on the knowledge base (i-iii), (Under Example 5, Table 1 and Definition 9) we have following result:

1.	$((a\&b) \Rightarrow c), v_{27})$	(i)
2.	(a, v_{24})	(iii)
3.	(b, v_{25})	(ii)
4.	$(\bot \nabla ((\top\&b) \Rightarrow c), v_{24} \otimes v_{27})$	(r_R on 1, 2)
4a.	$((b \Rightarrow c), v_{22})$	
5.	$(\bot \nabla (\top \Rightarrow c), v_{25} \otimes v_{22})$	(r_R on 3, 4a)
5a.	$(c, v_{18}) \equiv (c, MPTrue)$	

Therefore, the truth value of the sentence "Mary will be a good employee" is $(emp(Mary; good), MPTrue)))$, which means Mary will be a good employee is More Possibly True.

5. Conclusion

With the studies on finite monotonous hedge algebra as the linguistic truth value domain, the linguistic truth value domain is finite and the linearly order organized elements can act as base value set for truth domain of linguistic many-valued logic system. Then we was presented the formal system for linguistic many-valued logic based on resolution. Based on r_R, $r_{s\nabla}$, $r_{s\Rightarrow}$ and hedge moving rules, we build a deduction procedure and use it to solve the language deduction problem. In the future work, we will expand r_{MP} with linguistic modifiers for linguistic reasoning based on resolution.

Acknowledgement

The authors would like to express deep gratitude to Vietnam's National Foundation for Science and Technology Development (NAFOSTED) for its support.

References

[1] L. A. Zadeh, Fuzzy sets,Information and Control **8(3)**, 1965. p. 338-353.

[2] L. A. Zadeh, The concept of a linguistic variable and its application in approximate reasoning, Information Sciences, 1975. Part I – 8:199-249, Part II – 8:301-357, Part III – 9:43-80.

[3] Nguyen, C.H., Wechler, W., Hedge Algebras: An Algebraic Approach to Structure of Sets of Linguistic Truth Values, Fuzzy Sets and Syst. **35**, 281-293 (1990).

[4] Nguyen Cat Ho, Tran Dinh Khang, Huynh Van Nam, Nguyen HaiChau, Hedge Algebras, Linguistic-Valued Logic and their Applications to Fuzzy Reasoning, Intern. Journal of Fuzziness, Uncertainty and Knowledge-Based Systems, Vol. **7**, No. 4, August 1999, p. 347-361.

[5] Dinh Khac Dung, Steffen Hoelldobler, Tran Dinh Khang, The Fuzzy Linguistic Description Logic ALCFL, Proceedings of the Eleventh International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), 2006, pages 2096-2103.

[6] Van Hung Le, Fei Liu, Tran Dinh Khang, Linguistic logic programming and its applications, J. Theory and Practice of Logic Programming **9(3)** (2009) Cambridge University Press, 309-341.

[7] Tran Dinh Khang, Ta Quang Trung, Le Anh Phuong, The inverse mapping of hedges, Journal of Computer Science and Cybernetics, T. **26**, S.2 (2010), 119-129 (In Vietnamese).

[8] Tran Dinh Khang, Rules of Moving Hedges and the Subsumption Property, Journal of Computer Science and Cybernetics, T. **24**, (2) (2008), 97-106 (In Vietnamese).

[9] Le Anh Phuong, Tran Dinh Khang, A deductive Method in Linguistic Reasoning, Proceeding of IEEE 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, (2012) p 137-140.

[10] Novák, V., Perfilieva, I., Mockor, J. Mathematical principles of fuzzy logic, Kluwer Academic Publishers, (1999).

[11] J. Pavelka, "On fuzzy logic I: Many-valued rules of inference, II: Enriched residuated lattices and semantics of propositional calculi, III: Semantical completeness of some many-valued propositional calculi", Zeitschr. F. Math. Logik und Grundlagend. Math., **25** (1979) 45-52, 119-134, 447-464.

[12] Hsing-Tai Chung, Daniel G. Schwartz, A Resolution Based System for Sumbolic Approximate Reasoning, International Journal of Approximate Reasoning, **13** (1995); 201-246.

[13] Habiballa, H. Novak, V. Fuzzy General Resolution, Pro. Of Intl. Aplimat 2002. Bratislava, Slovak Technica University, 2002. 199-206.

[14] Lui Jun, Song Zhenming, Qin Keyun, A Resolution Procedure Based on a Fuzzy logic, Proceeding of The Ninth IEEE International Conference on Fuzzy Systems, (2000) 191-196.

[15] Yang Xu, Da Ruan, Etienne E. Kerre, Jun Liu,Alpha-Resolution principle based on lattice-valued propositional logic LP(X). Inf. Sci. **130(1-4)**, (2000) 195-223.

Le Anh Phuong received his Master of Computer Science from School of Information & Communication Technology, Hanoi University of Science and Technology, Vietnam in the year 2001 and B.Sc (with Honors) degree in Mathematic and Computer Science from Hue University of Education in 1996. He is now PhD student at School of Information & Communication Technology, Hanoi University of Science and Technology, Vietnam. He has contributed 10 journal and conference papers. His interest includes linguistic logic, uncertainty reasoning, computational logic.

Tran Dinh Khang received his Diplom in Mathematical Cybernetics and Computing Techniques, Technische Universitaet Dresden, Germany in the year 1986 and PhD in Computer Science, Hanoi University of Science and Technology, Vietnam, in 1999. He is now Associate Professor and Vice Dean of School of Information & Communication Technology, Hanoi University of Science and Technology. He has contributed about 60 journal and conference papers. His interest includes fuzzy logic and applications, computational logic, decision support systems.