ISSN 2047-3338

A Method for Linguistic Reasoning Based on Linguistic Lukaseiwicz Algebra

Le Anh Phuong[1] and Tran Dinh Khang[2]
[1]Department of Computer Science, Hue University of Education, Vietnam
[2]School of Information & Communication Technology, Hanoi University of Science and Technology, Vietnam
[1]leanhphuong@dhsphue.edu.vn, [2]khangtd@soict.hut.edu.vn

***Abstract*– This paper studies the linguistic truth value domain (AX) based on finite monotonous hedge algebra and then we extend lukaseiwicz algebra on [0,1] to linguistic lukaseiwicz algebra on linguistic truth value domain (AX), in an attempt to propose a derivatives system based on hedge moving rules and linguistic lukaseiwicz algebra for linguistic reasoning.**

***Index Terms*– Hedge Algebra, Linguistic Truth Value Domain, Linguistic Lukaseiwicz Algebra and Derivatives**

I. INTRODUCTION

INFORMATION science has brought about an effective tool to help people engaged in computing and reasoning based on natural language. The question is how to model the information processing of human? A method of computation with words (CWW) has been studied by Zadeh [1], [2], with the construction of the fuzzy set representing the concept of language and the reasoning based on the membership function. In [3] N. C. Ho, Wechler, W. proposed hedge algebraic structures in order to model the linguistic truth value domain. Based on the hedge algebraic structures, N.C. Ho et al [4] gave a method of linguistic reasoning, but also posed further problems to solve.

A method of direct reasoning was introduced by Zheng Pei [7]. Based on the domain of truth values $T = H \times Tr$, Zheng Pei used Lukasiewics algebra for the truth value domain T to solve the reasoning problem. This method proved to be simple, but it just examines a hedge that has impacts on the generators *True (False).*

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 algebraic algebra, one can build inverse mapping of hedges with limited length [8], allowing the expansion of hedge moving rules.

The writing suggests finite monotonous hedge algebra be the linguistic truth value domain, using hedge moving rules, hedge inverse mapping and linguistic lukasiewicz algebra as derivative system to solve the problem of reasoning. This paper is extend the result in [11].

The paper consists of five parts, the preliminaries is followed by section 2 presenting basic knowledge serving as theoretical foundation for the research. Section 3 is for research in the linguistic truth value domain based on hedge moving rules, hedge inverse mapping and linguistic lukaseiwicz algebra, Section 4 shows a derivative system on the linguistic truth value domain based on linguistic lukaseiwicz algebra to construct a procedure of reasoning. The last section is the conclusion.

II. 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]-[11].

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

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 $X = \{\delta c \mid c \in G, \delta \in H*\}$. 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 $AX = (X, G, H, >)$.

Each hedge $h \in H$ can be regarded as a unary function h: $X \rightarrow X; x \mapsto hx$. 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, σ∈ H−. In the hedge chain h_p... 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 ∀x∈X: hx > x iff kx < x;*

ii) h and k are compatible *if ∀x∈X: hx > x iff kx > x;*

iii) h modifies terms stronger or equal than k, denoted by h ≥k, if ∀x∈X

iv) (hx ≤ kx ≤ x) or (hx ≥ kx ≥x); h > k if h ≥ k and h ≠k;

v) h is positive w.r.t. *k if ∀x∈ X: (hkx<kx< x) or (hkx>kx> x);*

vi) h is negative w.r.t. *k if ∀x∈ X: (kx<hkx< x) or (kx>hkx> x).*

The most commonly used HAs are symmetric ones, in which there are exactly two generators, like e.g., *G = {True; False}*. In this paper, we only consider symmetric HAs. 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*∈*X*, let $H(x) = \{\sigma x | \sigma \in H^*\}$.

Definition 2: *An abstract algebra (X, G, H, >), where H ≠ ∅, 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 ≠ kand hx<kx then h'hx<k'kx, for all h, k, h', k' ∈H and x∈X.

(A5) If $u \notin H(v)$ *and u < v (u > v) then u <hv (u >hv, resp.), for any h ∈H.*

Example 1 Consider a *HA (X, {True; False}, H, >)*, where *H = {Very,More,Probably,Mol}*, 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* ∈ H^+ and *Mol* ∈ 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.

B. Inverse mapping of hedge

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

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

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*∪*{0,W,1}* of which 0, W, 1 are the smallest, neutral, and biggest element in *AX* respectively. A mapping $h^-: AX \longrightarrow AX$ 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) $x \leq y \Longrightarrow h^-(x) \leq h^-(y)$ 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:

$$(h_k h_{k-1} \dots h_1)^-(\delta c) = h_k^-(\dots(h_1^-(\delta c) \dots)$$

Then the rule (*RT*2) is generalized as follows:

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

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

III. LINGUISTIC TRUTH VALUE DOMAIN

A. 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 $\leq L$. Based on monotonous hedge algebra Mono – HA, we set finite monotonous hedge algebra to make linguistic truth value domain.

Definition 7: $(L - Mono - HA)$ $L - Mono - HA$, L is a natural number, is a Mono – HA with standard presentation of all elements having the length not exceeding L+1.

Definition 8: (*Linguistic truth value domain*) A linguistic truth value domain AX taken from a$L - Mono - HA = (X, \{c^+, c^-\}, H, \leq)$ is defined as AX=X∪{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 $2 - Mono - HA = (X, \{c^+, c^-\}, \{V, M, P\}, \leq)$(*V=Very; M=More; P=Possibly)*
$(P \in H^-, M, V \in H^{\mp}, M < V)$.

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+, Vc+, VVc+, 1*}.

Propositions 1 If we have $L - Mono - HA = (X, \{c^+, c^-\}, H, \leq)$, 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* hedges).

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

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, ..., h−1}, H+ = {h1, ..., hp}
  x = lk l ... l1 c  with  c ∈ {T,F},  k ≤ L
Output: Finding index so that v_index = x
Methods:
```

$$M = 3 + 2 * \sum_{i=0}^{L}(p+q)^i$$

```
if x=0 then index=1 endif
if x=W then index=(M+1)/2 endif
if x=1 then index=M endif
index = (M+1)/2 + 1 + qAX^1
fori=1 to k-1 do
{ find j such that l_i = h_j
   if j>0 then index =index+(j−1)|AX^i| + q|AX^(i+1)| + 1;
   if j<0 then index = index-(|j| − 1)|AX^i| − p|AX^(i+1)| − 1; }
endfor
find j such that l_k = h_j  /*j > 0 then l_k ∈ H+, else l_k ∈ H−*/
if k<L then
{  if j>0 then index =index+(j − 1)|AX^k| + q|AX^(k+1)| + 1;
   if j<0 then index = index-(|j| − 1)|AX^k| − p|AX^(k+1)| − 1; }
Else index =index+j;
if c=False then index = (M+1) - index
return (index)
```

$\{*|AX^i| = \sum_{k=0}^{L-i}(p+q)^k *\}$

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

Table 1: Inverse mapping of hedges

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

B. *Linguistic lukaseiwicz algebra*

In logic, the truth value domain is shown by a algebra structure with calculations ∧,∨,′,→. Many valued logic has finite truth value domain including elements ardomaind according to linear order on [0,1] and Lukasiewicz algebra is an algebra structure for this truth value domain.

Definition 8: [10] The structure $\mathcal{L} = ([0,1], \wedge, \vee, \otimes, ', \to, 0, 1)$ 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 ∧,∨,⊗,′,→ are operators defined as follows:

i) $a \wedge b = min(a,b)$
ii) $a \vee b = max(a,b)$
iii) $a \to b = min(1, 1 - a + b)$
iv) $a' = 1 - a$
v) $a \otimes b = max(0, a + b - 1)$

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

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

Definition 9: The structure $\mathcal{L}_n = (\text{AX}, \wedge, \vee, \otimes, ', \to, 0, 1)$ 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 $\wedge, \vee, \otimes, ', \to$ are operators defined as follows:

i) $v_i \vee v_j = v_{max\{i,j\}}$
ii) $v_i \wedge v_j = v_{min\{i,j\}}$
iii) $(v_i)' = v_{n-i+1}$
iv) $v_i \to_L v_j = v_{min\{n, n-i+j\}}$
v) $v_i \otimes v_j = v_1 \vee v_{i+j-n}$

Next, we can using the linguistic Lukasiewicz algebra with AX of deductive methods for linguistic reasoning.

IV. DERIVED SYSTEM ON LINGUISTIC LUKASEIWICZ ALGEBRA

A. Derived system

One vague sentence can be represented by *p(x;u)*, herein *x* is a variable, *u* is a vague concept. In general, by an assertion is one pair A=(*p(x;u), δc*) (Symbol: *(P,v)*), herein *p(x;u)* is a vague sentence, δc is a linguistic truth value. One knowledge base *K* is a finite set of assertions. From the given knowledge base *K*, we can deduce new assertions by using on derived rules. In [4-6,11], the hedge moving rules are set:

RT1: $\dfrac{(p(x;hu),\delta c)}{(p(x;u),\delta hc)}$ GRT2: $\dfrac{(p(x;u),\delta c)}{(p(x;hu),h^{-}(\delta c))}$

From propositional calculus point of view, rules $\wedge$, $\vee$, $\longrightarrow$ can be extended for the linguistic truth value domain. If using operations $\wedge$, $\vee$, ‘, $\longrightarrow$ as links in Linguistic lukasiewicz algebra for the linguistic truth values domain, we have following $\wedge$, $\vee$, $\longrightarrow$ rules:

R1: $\dfrac{(p(x;u),v_i)\ and\ (q(y;v),v_j)}{(p(x,u) \wedge q(y,v), v_i \wedge v_j)}$

R2: $\dfrac{(p(x;u),v_i)\ or\ (q(y;v),v_j)}{(p(x,u) \vee q(y,v), v_i \vee v_j)}$

R3: $\dfrac{(p(x;u),v_i) \to (q(y;v),v_j)}{(p(x;u) \to q(y,v), v_i \to v_j)}$

R4: $\dfrac{(p(x;u) \to q(y,v),v_i),(p(x;u),v_j)}{(q(y,v), v_i \otimes v_j)}$

R5: $\dfrac{(p(x;u),v_i) \to (q(y;v),v_j),(p(x;u),v_k)}{(q(y,v),(v_i \to v_j) \otimes v_k)}$

RS: $\dfrac{(p(x;u),\delta c)}{(p(a;u),\delta c)}$; RE: $\dfrac{P \leftrightarrow Q,(F(P),\delta c)}{F(Q/P),\delta c)}$

Herein, R5 is an extension of R4.

Given $\alpha, \beta, \delta, \theta, \partial, \alpha', \beta', \delta', \partial'$ is the hedge strings. Get $\alpha = h_1 h_2 \dots h_k$, symbol $\alpha^{-1} = h_k h_{k-1} \dots h_1$. We have following propositions:

Proposition 2:

$$\dfrac{\begin{array}{l}(p(x;\delta u) \to q(y;\partial v), \alpha c) \\ (p(x;\delta' u), \alpha' c)\end{array}}{(q(y;\partial v),(\delta^{-}(\alpha' \delta'^{-1} c) \otimes \alpha c))}$$

Proof

According to RT1 we have: $(p(x;u), \alpha' \delta'^{-1} c)$;

Then, applying GRT2 we have: $(p(x;\delta u), \delta^{-}(\alpha' \delta'^{-1} c))$;

Finally, using R4 we have:

$$(q(y;\partial v),(\delta^{-}(\alpha' \delta'^{-1} c) \otimes \alpha c)) \ \blacksquare$$

Proposition 3:

$$\dfrac{\begin{array}{l}(p(x;\delta u) \wedge q(y;\partial v) \to r(z,\theta t)), \alpha c) \\ (p(x;\delta' u), \alpha' c) \\ (q(y;\partial' v), \beta' c)\end{array}}{(r(z;\theta t),(\delta^{-}(\alpha' \delta'^{-1} c) \wedge \partial^{-}(\beta' \partial'^{-1} c)) \otimes \alpha c))}$$

Proof

According to RT1 we have:

$$(p(x;u), \alpha' \delta'^{-1} c); (q(y;v), \beta' \partial'^{-1} c);$$

Then, using GRT2 we have:

$$(p(x;\delta u), \delta^{-}(\alpha' \delta'^{-1} c)); (q(y;\partial v), \partial^{-}(\beta \partial'^{-1} c));$$

Next, with R1 we have:

$$(p(x;\delta u) \wedge q(y;\partial v), \delta^{-}(\alpha' \delta'^{-1} c) \wedge \partial^{-}(\beta \partial'^{-1} c));$$

Finally, using R4 we have:

$$(r(z;\theta t),(\delta^{-}(\alpha' \delta'^{-1} c) \wedge \partial^{-}(\beta' \partial'^{-1} c)) \otimes \alpha c)) \ \blacksquare$$

Proposition 4:

$$\dfrac{\begin{array}{l}(p(x;\delta u) \vee q(y;\partial v) \to r(z,\theta t)), \alpha c) \\ (p(x;\delta' u), \alpha' c) \\ (q(y;\partial' v), \beta' c)\end{array}}{(r(z;\theta t),(\delta^{-}(\alpha' \delta'^{-1} c) \vee \partial^{-}(\beta' \partial'^{-1} c)) \otimes \alpha c))}$$

Proof

According to RT1 we have:

$$(p(x;u), \alpha' \delta'^{-1} c); (q(y;v), \beta' \partial'^{-1} c);$$

Then, using GRT2 we have:

$$(p(x;\delta u), \delta^{-}(\alpha' \delta'^{-1} c)); (q(y;\partial v), \partial^{-}(\beta \partial'^{-1} c));$$

Next, using R2 we have:

$$(p(x;\delta u) \vee q(y;\partial v), \delta^{-}(\alpha' \delta'^{-1} c) \vee \partial^{-}(\beta \partial'^{-1} c));$$

Finally, applying R4 we have:

$$(r(z;\theta t),(\delta^{-}(\alpha' \delta'^{-1} c) \vee \partial^{-}(\beta' \partial'^{-1} c)) \otimes \alpha c)) \ \blacksquare$$

Proposition 5:

$$\dfrac{\begin{array}{l}(p(x;\delta u), \alpha c) \to (q(y;\partial v), \beta c) \\ (p(x;\delta' u), \alpha' c)\end{array}}{(q(y;\partial' v), \partial'^{-}((\alpha \delta^{-1} c \to \beta \partial^{-1} c) \otimes \alpha' \delta'^{-1} c))}$$

Proof:

Applying RT1 we have:

$$(p(x;u), \alpha \delta^{-1} c); (q(y;v), \beta \partial^{-1} c); (p(x;u), \alpha' \delta'^{-1} c);$$

Then, using R5 we have:

$$(q(y;v),(\alpha \delta^{-1} c \to \beta \partial^{-1} c) \otimes \alpha' \delta'^{-1} c);$$

Finally, using GRT2 we have:

$$(q(y;\partial' v), \partial'^{-}((\alpha \delta^{-1} c \to \beta \partial^{-1} c) \otimes \alpha' \delta'^{-1} c)) \ \blacksquare$$

B. 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 conclutions: $C(K) = \{(P, v): K \vdash (P, v)\}$. 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 based on deriveds and Proposition (2-5)

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

Algorithm 2 (*Deductive procedure*)

Input: Knowledge base set K; $L - Mono - HA$

Output: Truth value of the clause (P, v)

Method:

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: Finding the truth value expression of the conclusion using Proposition (2-5);

Step 3: Transfering the truth value δc in the expression found in Step 2 into v_i: $v_i = \delta c$ (Algorithm 1)

Step 4: Calculating the truth value expression based on Lukasiewicz calculations and application inverse of Hedge;

Step 5: Making the truth value of conclusion clause.

C. Examples

Example 6 Given the following knowledge base:

i) If a student studying *more hard* and his university is *very high-raking*, then he will be a *good employee* is *possibly 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:

1.(studying(x;MHard) $\wedge$ is(Univ(x);VHi_ra) $\rightarrow$ emp(x;good),PVTrue))

(Base on the hypothesis(i))

2. (is(Univ(Mary); VHi_ra), PTrue))

(Base on (ii))

3. (studying(Mary; VHard), MTrue)

(Base on (iii))

Based on the knowledge base (i-iii) and Proposition 3, we have following result:

(emp(x;good), ($\boldsymbol{M^{-}(MVTrue)} \wedge \boldsymbol{V^{-}(PVTrue)}$ $\otimes$PVTru$\boldsymbol{e}$)

We have calculations**:** **(**Under Example 5, Table 1 and Lukasiewicz operations defined in Part 2 and Part 3)

$$M^{-}(MVTrue) = MMTrue = v_{23}$$

$$V^{-}(PVTrue) = True = v_{20}$$

$$PVTrue = v_{25}$$

$$M^{-}(MVTrue) \wedge V^{-}(PVTrue) = v_{23} \wedge v_{20} = v_{20}$$

$(M^{-}(MVTrue) \wedge V^{-}(PVTrue) \otimes$ PVTru$e = v_{20} \otimes v_{25} = v_{16} = PPTrue$

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

V. CONCLUSION

With the studies on finite monotonous hedge algebra as the linguistic truth value domain (AX), the linguistic truth value domain (AX) is finite and the linear order organized elements can act as base value set for truth domain of logic system. Based on derived system on linguistic lukaseiwicz algebra, we build an deduction procedure and use it to solve the linguistic deduction problem.

ACKNOWLEDGMENT

This work is partially supported by Vietnam's National Foundation for Science and Technology Development (NAFOSTED).

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 Hai Chau**,** Hedge Algebras, Linguistic-Valued Logic and their Applications to Fuzzy Reasoning, *Intern. Journal of Fuzziness, Uncertainty and Knowl. 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 ALC_{FL}, *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] Zheng Pei**,** The Algebraic Properties of Linguistic Value "Truth" and Its Reasoning**,** *P. Melin et al, (Eds): IFSA 2007*, LNAI 4529, pp 436-444, 2007

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

[9] 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)

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

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

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 (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 5 technical papers in International Journals and Conferences. 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 1990 and PhD in Computer Science at School of Information & Communication Technology, Hanoi University of Science and Technology, Vietnam in 1999. He is now Prof and Vice Dean of School of Information & Communication Technology, Hanoi University of Science and Technology. He has contributed 30 technical papers in International Journals and Conferences His interest includes Fuzzy logic and applications, Computational logic, Decision support systems.