IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012
ISSN (Online): 1694-0814
www.IJCSI.org

Generalized If ... Then ... Else Inference Rules with Linguistic Modifiers for Approximate Reasoning

Le Anh Phuong[1], 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

Abstract

In this paper, based on the our previous researchs about generalized modus ponens with linguistic modifiers for If … Then rules, we propose generalized If…Then…Else inference rules with linguistic modifiers in linguistic many–valued logic framework with using hedge moving rules for approximate reasoning.

Keywords: *Hedge algebra, linguistic truth value domain, generalized If…Then…Else inference rules, linguistic many–valued logic, approximate reasoning.*

1. 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 (HA) structures, N.C. Ho et al [4] gave a method of linguistic reasoning, but also posed further problems to solve.

Studied in [10-15] are generalized If…Then…Else inference rules for approximate reasoning with fuzzy conditional proposition based on membership function of fuzzy sets.

In [9,10], we studied generalized modus ponens with linguistic modifiers for If…Then rules in linguistic many valued-logic and its application for approximate reasing.

In this paper, following our previous works, we are studing generalized If…Then…Else inference rules with linguistic modifiers in linguistic many-valued logic, with using hedge moving rules and hedge inverse mapping 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 shows the generalized If…Then…Else inference rules in linguistic many–valued logic framework with using hedge moving rules for approximate 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 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 HA = $(X, G, H, >)$.

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,* $\sigma \in H-$. *In the hedge chain* $h_p \ldots h_1$, 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}*, 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:

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

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.

3. Linguistic many-valued logic

3.1 Lingusitic truth valued domain

In real life, people only use a string of hedge with finite length for an 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 exceed 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+, Pc+, MPc+, VPc+, c+, PMc+, Mc+, MMc+, VMc+, PVc+, Vc+, MVc+, VVc+, 1*}.

Propositions 1 If we have $L-Mono-HA=(X,\{c^{+},c^{-}\},H,\leq)$, the linguistic truth value domain AX is finite to 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,

$H^{-}=\{h_{-q},\dots,h_{-1}\},H^{+}=\{h_{1},\dots,h_{p}\}$

$x=l_{k}l\dots l_{1}c$ with $c\in\{T,F\},\ k\leq L$

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;
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)|AX^i| + q|AX^{i+1}| + 1;
   if j<0 then index = index-(|j|-1)|AX^i| - p|AX^{i+1}| - 1;
}

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 $2-Mono-HA=(X,\{c^{+},c^{-}\},\{V,M,\ P\},\leq)$: ($k\in H$) (see Table 1)

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^{+}	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

3.2 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 $AX=\{v_{i},i=1,2,\dots,n\}$ with $v_{1}=0$ and $v_{n}=1$ in finite monotonous hedge algebra and linear order or $AX=\{v_{i},i=1,2,\dots,n;\ v_{1}=0,v_{n}=1\ \ and\ \forall\ 1\leq i,j\leq n\colon v_{i}\geq\ v_{j}\Leftrightarrow i\geq j\}$.

In linguistic many–valued logic, the truth degree of proposition is $v_{i}\in AX$.

T-norms, T-conorm, implicators and negation operator are used as in Fuzzy logic. In many-valued logic, the aggregation functions of lukasiewicz are often used. In this context and with N truth degrees, they are difined by [9-11]:

$$T_{L}(v_{i},v_{j})=v_{max(1,i+j-N)}$$
$$S_{L}(v_{i},v_{j})=v_{min(N,i+j+1)}$$
$$I_{L}(v_{i},v_{j})=v_{min(N,N-i+j)}$$
$$N(v_{i})=v_{N-i+1}$$
$$v_{i}\wedge v_{j}=v_{\min(i,j)}$$
$$v_{i}\vee v_{j}=v_{\max(i,j)}$$

We can use T-norms, T-conorm, implicators, ∧, ∨ and negation operator above in linguistic many-valed logic with $v_i, v_j \in AX$.

4. Generalized If…Then…Else inference rules with linguistic modifiers

4.1 Generalized If…Then…Else rules

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 [4,5,6], the hedge moving rules are set:

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

And

(p(x,NOT(u)), δc) ⇔(p(x,u),NOT(δc)) (RN)

(p(x,NOT(δu))⇔(p(x, δ(NOT(u))) (RNH)

(with δ is the hedge string)

In [9,10], the generalized modus ponens was proposed

GMP: $\frac{(p(x;u)\to q(y,v),v_i),(p(x;u),v_j)}{(q(y,v),T_L(v_i,v_j)}$

EGMP: $\frac{(p(x;u),v_i)\to(q(y;v),v_j),(p(x;u),v_k)}{(q(y,v),T_L(I_L(v_i,v_j),v_k)}$

Herein, EGMP is an extension of EGMP.

From GMP, EGMP and RN, we have:

NGMP: $\frac{(p(x;NOT(u))\to q(y,v),v_i),(p(x;u),v_j)}{(q(y,v),T_L(v_i,N(v_j)))}$

NEGMP: $\frac{(p(x;NOT(u)),v_i)\to(q(y;v),v_j),(p(x;u),v_k)}{(q(y,v),T_L(I_L(v_i,v_j),N(v_k))}$

Let us consider the fuzzy inference in which the fuzzy conditional "If…Then…Else" is contained:

(ITE):

Antecedent 1: *If X is A Then Y is B Else Y is C*
Antecedent 2: *X is A'*

Conclusion: *Y is D*

Arcording to [12,13], "If-Then-Else" inferences rules may be a quite natural demand:

Antecedent 1: *If X is A Then Y is B Else Y is C*
Antecedent 2: *X is A*

Conclusion: *Y is B*

Antecedent 1: *If X is A Then Y is B Else Y is C*
Antecedent 1: *X is NOT A*

Conclusion: *Y is C*

When, "If-Then-Else" inferences rules can be divided into two subschemes, that is:

(IT1):

Antecedent 1: *If X is A Then Y is B*
Antecedent 2: *X is A'*

Conclusion: *Y is B'*

(IT2):

Antecedent 1: *If X is NOT A Then Y is C*
Antecedent 1: *X is A'*

Conclusion: *Y is C'*

According to [12,13], the conclusion "Y is D" in the "If…Then…Else" inference scheme, should be "Y is B'" and "Y is C'", that mean:

If X is A Then Y is B Else Y is C ⇔ If X is A Then Y is B AND If X is NOT A Then Y is C

Therefore, the conclusion D in (ITE) is computed by aggregation Min of B' in (IT1) and C' in (IT2).

We can generalized (IT1) and (IT2) with linguistic modifiers following:

Given $\alpha,\beta,\delta,\theta,\partial,\alpha',\beta',\delta',\partial'$ is the hedge strings. Get $\alpha = h_1h_2\dots h_k$, symbol $\alpha^{-1} = h_kh_{k-1}\dots h_1$. We have following propositions (*Generalized If-Then-Else inference rules with linguistic modifiers)*:

Proposition 2

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

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 GMP we have:

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

Proposition 3

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

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 EGMP we have:

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

Finally, using GRT2 we have:

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

Proposition 4

$$\frac{\begin{array}{l}(p(x;NOT(\delta u))\to q(y;\partial v),\alpha c)\\(p(x;\delta' u),\alpha' c)\end{array}}{(q(y;\partial v),T_L(\alpha c,N(\delta^{-}(\alpha'\delta'^{-1}c)))}$$

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 NGMP we have:

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

Proposition 5

$$\frac{\begin{array}{l}(p(x;NOT(\delta u)),\alpha c)\to(q(y;\partial v),\beta c)\\(p(x;\delta' u),\alpha' c)\end{array}}{(q(y;\partial' v),\partial'^{-}(T_L(I_L(\alpha\delta^{-1}c,\beta\partial^{-1}c),N(\alpha'^{\delta'^{-1}}c))))}$$

Proof:

Application RNH we have:

$$p(x;NOT(\delta u)) \Leftrightarrow p(x;\delta(NOT(u)))$$

Applying RT1 we have:

$$(p(x;NOT(u)),\alpha\delta^{-1}c);$$

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

Then, using NEGMP we have:

$$(q(y;v),T_L(I_L(\alpha\delta^{-1}c,\beta\partial^{-1}c),N\left(\alpha'^{\delta'^{-1}}c\right)));$$

Finally, using GRT2 we have:

$$(q(y;\partial'v),\partial'^{-}(T_L(I_L(\alpha\delta^{-1}c,\beta\partial^{-1}c),N\left(\alpha'^{\delta'^{-1}}c\right)))) \blacksquare$$

4.2 Deductive procedure besed on generalized "If…Then…Else" inference rules

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 *(┐P,v)*.

Here, we build an deduction procedure (Algorithm 2) based on hedge moving rules and Proposition (2-5) for solving approximate reasoning.

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 = δc (Algorithm 1)
Step 4: Calculating the truth value
        expression based on T-norms, T-
        conorm, implicators, negate and
        ∧ (Min) operation was defined
        above an application inverse of
        hedge;
Step 5: Making the truth value of
        conclusion clause.
```

4.3 Examples

Example 6 Given the following knowledge base:

i) If a student studying *more hard* Then he will be a *good employee* is *possibly very true* Else he will be a *good employee* is *possibly very false*

ii) Mary is studying *very hard* is *more true*.

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

We can be divided into two subschemes:

(IT1):

i) If a student studying *more hard* then he will be a *good employee* is *possibly very true.*

ii) Mary is studying *very hard* is *more true*.

(IT2):

i) If a student studying *NOT(more hard)* then he will be a *good employee* is *possibly very false*

ii) Mary is studying *very hard* is *more true*.

(IT1):

By formalizing. (i) – (ii) an be rewritten by follow:

1. (studying(x;MHard) → emp(x;good),*PVTrue*)) (Base on the hypothesis(i))

2. (studying(Mary; VHard), *MTrue*) (Base on (ii))

Based on the knowledge base (i-ii) and Proposition 2, we have following result:

(emp(x;good), $T_L(PVTrue, M^-(MVTrue))$)

We have calculations**: (**Under Example 5, Table 1 and T-norms, T-conorm and implicators defined in Part 3)

$$M^-(MVTrue) = VMTrue = v_{24}$$
$$PVTrue = v_{25}$$
$$T_L(PVTrue, M^-(MVTrue)) = T_L(v_{25,}v_{24}) = v_{20}$$

(IT2):

By formalizing. (i) – (ii) an be rewritten by follow:

1. (studying(x;NOT(MHard)) → emp(x;good),*PVFalse*)) (Base on the hypothesis(i))

2. (studying(Mary; VHard), *MTrue*) (Base on (ii))

Based on the knowledge base (i-ii) and Proposition 4, we have following result:

(emp(x;good), $T_L(PVTrue, N(M^-(MVTrue)))$)

We have calculations**: (**Under Example 5, Table 1 and T-norms, T-conorm, implicators and negation operator defined in Part 3)

$$N(M^-(MVTrue)) = N(VMTrue) = N(v_{24}) = v_6$$
$$PVTrue = v_{25}$$
$$T_L(PVTrue, M^-(MVTrue)) = T_L(v_{25,}v_6) = v_2$$

According to (IT1) and (IT2), we have the truth value of the sentence "*Mary will be a good employee*" is (emp(Mary; good), $v_{20} \wedge v_2 = v_2 = VVFalse$), which means Mary will be a good employee is **Very Very False**.

Example 7 Given the following knowledge base:

i) If a student studying *more hard is possibly true* Then he will be a *good employee* is *possibly very true* Else Then he will be a *good employee* is *possibly very false*

ii) Mary is studying *very hard* is *more true*.

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

(IT1):

i) If a student studying *more hard is possibly true* Then he will be a *good employee* is *possibly very true*

ii) Mary is studying *very hard* is *more true*.

(IT2):

i) If a student studying *NOT(more hard) is possibly true* Then he will be a *good employee* is *possibly very false*

ii)Mary is studying *very hard* is *more true*.

(IT1)

By formalizing. (i) – (iii) an be rewritten by follow:

1. (studying(x;MHard),*PTrue*) → emp(x;good),*PVTrue*)) (Base on the hypothesis(i))

2. (studying(Mary; VHard), *MTrue*) (Base on (ii))

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

(emp(x;good),$T_L(I_L(PMTrue, PVTrue), MVTrue)$)

We have calculations: (Under Example 5, Table 1 and T-norms, T-conorm and implicators defined in Part 3)

$$PMTrue = v_{21};\ PVTrue = v_{25}; MVtrue = v_{27}$$
$$I_L(PMTrue, PVTrue) = I_L(v_{21}, v_{25}) = v_{29}$$
$$T_L(v_{29}, v_{27}) = v_{27}$$

(IT2)

By formalizing. (i) – (iii) an be rewritten by follow:

1. (studying(x;NOT(MHard)),*PTrue*) → emp(x;good),*PVTrue*)) (Base on the hypothesis(i))

2. (studying(Mary; VHard), *MTrue*) (Base on (ii))

Based on the knowledge base (i-ii) and Proposition 5, we have following result:

(emp(x;good),$T_L(I_L(PMTrue, PVTrue), MVTrue)$)

We have calculations: (Under Example 5, Table 1 and T-norms, T-conorm and implicators defined in Part 3)

$$PMTrue = v_{21};\ PVTrue = v_{25}$$
$$N(MVtrue) = N(v_{27}) = v_3$$
$$I_L(PMTrue, PVTrue) = I_L(v_{21}, v_{25}) = v_{29}$$
$$T_L(v_{29}, v_3) = v_3$$

According to (IT1) and (IT2), we have the truth value of the sentence "*Mary will be a good employee*" is (emp(Mary; good), $v_{27} \wedge v_3 = v_3 = MVFalse$), which means Mary will be a good employee is **More Very False**.

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 linear order organized elements can act as base value set for truth domain of logic system. In this paper, we studied generalized If…Then…Else inference rules with linguistic modifiers and building an deduction procedure and use it to solve the language deduction problem. In furture works, we would be to study an inference formalization for more complex rules.

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] 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)

[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*, (URKE 2012) p 137-140

[10] Le Anh Phuong, Tran Dinh Khang, Linguistic Reasoning based on Generalized Modus Ponens with Linguistic Modifiers and Hedge Moving Rules, *Proceedings of 2012 International Conference on Fuzzy Theory and Its Application* (iFUZZY 2012) (Accepted)

[11] Saossen Bel Hadj Kacem, Amel Borgi. Khaled Ghedira, Generalized Modus Ponens based on linguistic modifiers in symbolic multi-valued frameword, *Proceeding of the 38th International Symposium on Multiple Valued Logic*, (2008) p150 – 155

[12] D. Ruan, E.E. Kerre, On If–Then-Else Inference Rules, *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, October 14-17, Beijing, China, vol.2 of 4, (1996) 1420-1425

[13] Masaharu Mizumoto, Fuzzy inference methods with "IF…THEN…ELSE…", *Workshop on Fuzzy sets and Management Science*, Brussels (1981)

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 10 technical papers in journals and conferences. His interest includes linguistic many-valued logic, fuzzy 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 60 technical papers in journals and conferences. His interest includes fuzzy logic and applications, computational logic, decision support systems.