Abstract

Based on our previous researchs about generalized modus ponens (GMP) with linguistic modifiers for If … Then rules, this paper proposes new generalized modus tollens (GMT) inference rules with linguistic modifiers in linguistic many–valued logic framework with using hedge moving rules for inverse approximate reasoning.

Keywords: Hedge algebra, linguistic truth value domain, generalized modus tollens inference rules, linguistic many–valued logic, inverse 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 human’s information processing procedure? 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 approximate reasoning based on the membership function.

In [3] N. C. Ho, Wechler, W. proposed hedge algebraic (HA) structures in order to model the linguistic truth value domain. Based on the hedge algebraic structures, N.C. Ho et al [4] not only gave a method of linguistic reasoning, but also posed further problems to solve.

Reasoning is a mental activity that allows us to derive new premises from the given ones with some degree of confidence.

In a rule-based system, from a given rule (antecedent-consequent condition) and an observed state of antecedent, we conclude something by applying a method of inference which is called forward approximate reasoning (using generalized modus ponens for solving forward approximate reasoning). A problem with forward method is that many rules may be applicable for a particular observation (data on antecedent) as the whole process is not directed toward a goal.

<table>
<thead>
<tr>
<th>FOR ward APPROXIMATE REASONING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent 1: If X is A Then Y is B</td>
</tr>
<tr>
<td>Antecedent 2: X is A'</td>
</tr>
<tr>
<td>Conclusion: Y is B'</td>
</tr>
</tbody>
</table>

(With X, Y are linguistic variables; A and A*, B and B* are fuzzy sets)

In inverse approximate reasoning, the method works with a final state and is always directed toward the working memory for a goal. Thus, in a rule-based system, from a given rule (antecedent-consequent condition) and an observed state of consequent, we conclude something on the state of the antecedent by applying a method of inference which is called inverse approximate reasoning (using generalized modus tollens for solving inverse approximate reasoning).
The problem of inverse approximate reasoning was considered in [18]. The study focused on finding fuzzy data that would produce a conclusion. To deal with the method of inverse approximate reasoning, Papis, Eslami, Buckkley, Dieulot and Borne [19-23] investigated the solution of fuzzy relation equation and chose the best from the solution set. In [24-27], Revault d’allones studied the solution of fuzzy relation equation and chose the best solution from the set of linguistic values.

2.1 Monotinous 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 \( G = \{ \text{True; False} \} \) and a nonempty finite set \( H \) of hedges, the set \( X \) of linguistic values is \( \{ \text{c|c} \in G, \delta \in H^\ast \} \).

Furthermore, if we consider \( \text{True} > \text{False} \), then this order relation also holds for other pairs, e.g., \( \text{VeryTrue} > \text{MoreTrue} \). 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 \( X = \{ \text{c|c} \in G, \delta \in H^\ast \} \). 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 \( HA = (X, G, H, >, \cdot) \).

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 \( I \in H \) 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 \( \sigma \) is a word over \( H \), \( \sigma \in H^\ast \). In the hedge chain \( h_0... h_n \), \( h_0 \) is called the first hedge whereas \( h_n \) 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 \( \forall x \in X: (hx \leq kx \leq x) \text{ or } (hx \geq kx \geq x) \); \( h > k \) if \( h \geq k \) and \( h \neq k \);

iv) \( h \) is positive w.r.t. \( k \) if \( \forall x \in X: (hkx < kx < x) \text{ or } (hkx > kx > x) \);

v) \( h \) is negative w.r.t. \( k \) if \( \forall x \in X: (kx < hxkx < x) \text{ or } (kx > hx < x) \).

The most commonly used HA are symmetric ones, in which there are exactly two generators, like e.g., \( G = \{ \text{True; False} \} \). 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 \( H^+ = \{ h \in H | h c^+ > c^+ \} \) and \( H^- = \{ h \in H | h c^+ < c^+ \} \). For each value \( x \in X \), let \( H(x) = \{ \alpha x | \alpha \in H^+ \} \).
Definition 2

An abstract algebra \((X, G, H, >)\), where \(H \neq \emptyset\), \(G = \{c^+, c^-\}\) and \(X = \{r| r \in G, \sigma \in H\}\), 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 \(h \neq k\), \(h \leq k\) then \(h' \leq k'\), for all \(h, k, h', k' \in H\) and \(x \in X\).

(A5) If \(u \notin H(v)\) and \(u < v\) (\(u > v\) resp.), for any \(h \in H\).

Example 1

Consider a HA \((X, \{True, False\}, H, >)\), where \(H = \{Very, More, Probably, Mol\} \) (More or less).\)

(i) Very and More are positive wrt Very and More, and negative wrt Probably and Mol.

(ii) Probable 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 \(k \in H^-(H^+)\).

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 \(h, k \in H \cup \{I\}\), \(h \geq k\) 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 \(h \leq k\) and \(h \neq k\).

Example 2

The HA in example 1 is Mono-HA. The order relation \(>_h\) in \(H \cup \{I\}\), is \(Very > More > I > Probably > Mol\).

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:

\[
RT1: \frac{p(x;\alpha), \delta c}{(p(x;\alpha), \delta hc)} \quad RT2: \frac{p(x;\alpha), \delta hc}{(p(x;\alpha), \delta c)}
\]

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 “Very 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 \(AX = X \cup \{0, W, I\}\) of which 0, W, I are the smallest, neutral, and biggest element in \(AX\) respectively.

A mapping \(h^\prime: AX \rightarrow AX\) is called inverse mapping of \(h\) if it meets the following conditions:

i) \(h^\prime(\delta c) = \delta c\) of which \(c \in G = \{c^+, c^-\}\), \(\delta \in H^+\).

ii) \(x \leq y \Rightarrow h^\prime(x) \leq h^\prime(y)\) 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:

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

Then the rule (RT2) is generalized as follows:

\[
GRT2: \frac{p(x;\alpha), \delta c}{(p(x;\alpha), \delta c)}
\]

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 valued domain*

In real life, people only use a string of hedge with finite length for vague concepts or to make it not exceed $L$ – any positive number. In case that intellectual base has a value with length of hedge string is bigger than $L$, we need to approximate the value having hedge string $\leq L$. Based on monotonous hedge algebra $\text{Mono} - \text{HA}$, we set finite monotonous hedge algebra to make linguistic truth value domain.

**Definition 7** ($L - \text{Mono} - \text{HA}$)

$L - \text{Mono} = \text{HA}$, $L$ is a natural number, is a $\text{Mono} - \text{HA}$ 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 $L - \text{Mono} - \text{HA} = (X, \{c^+, c^-, c^0\}, V, M, P, \leq)$ of which $0, W, 1$ are the smallest, neutral, and biggest elements respectively in $AX$.

**Example 4**

Given finite monotonous hedge algebra $2 - \text{Mono} - \text{HA} = (X, \{c^+, c^-, c^0\}, V, M, P, \leq)$ of which $0, W, 1$ are the smallest, neutral, and biggest elements respectively in $AX$.

**Propositions 1**

If we have $L - \text{Mono} - \text{HA} = (X, \{c^+, c^-, c^0\}, V, M, P, \leq)$, the linguistic truth value domain $AX$ is finite to number of elements $|AX| = 3 + 2 \sum_{i=0}^{L-1} |H|^i$ 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; $i$ from $i=L$ we have $2n^L$ more elements.

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

According to the definition of linear order relation in monotonous hedge algebra $\text{Mono} - \text{HA}$, 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) $AX = \{v_1 = 0, v_2 = VVc^+, v_3 = MVc^+, v_4 = PC^+, v_5 = PMc^-, v_6 = PMc^+, v_7 = MMc^+, v_8 = M^+, v_9 = c^-, v_{10} = M^-, v_{11} = V^-, v_{12} = M^-, v_{13} = PC^-, v_{14} = PMc^-, v_{15} = W, v_{16} = PC^-, v_{17} = PC^+, v_{18} = PMc^+, v_{19} = PMc^+, v_{20} = c^+, v_{21} = M^+, v_{22} = M^+, v_{23} = MMc^+, v_{24} = VMC^+, v_{25} = VVc^+, v_{26} = Vc^+, v_{27} = M^+, v_{28} = V^+, v_{29} = 1\}$.

We can determine the index of $v$ by Algorithm 1:

**Algorithm 1** (*Finding index*)

**Input:** Domain (Truth) of $L - \text{Mono} - \text{HA}$ is $AX$,

$H^- = \{h_{-q}, \ldots, h_{-1}\}, H^+ = \{h_1, \ldots, h_p\}$

**Output:** Finding index so that $v_{\text{index}} = x$

**Methods:**

$\begin{align*}
M &= 3 + 2 \sum_{i=0}^{L-1} (p + q)^i \\
&\text{if } x=0 \text{ then } \text{index} = 1; \\
&\text{if } x=W \text{ then } \text{index} = (M+1)/2; \\
&\text{if } x=1 \text{ then } \text{index} = M; \\
&\text{index} = (M+1)/2 + 1 + qAX^1 \\
&\text{if } i=1 \text{ to } k-1 \text{ do} \\
&\text{find j such that } l_j = h_i \\
&\text{if } j>0 \text{ then } \text{index} = \text{index} + (j-1)|AX|^1 + q|AX|^{k+1} + 1; \\
&\text{if } j<0 \text{ then } \text{index} = \text{index} - (|j| - 1)|AX|^1 - p|AX|^{k+1} - 1; \\
&\text{find j such that } l_k = h_i / *j > 0 \text{ then } l_k \in H^+; \\
&\text{else } l_k \in H^- */ \\
&\text{if } k>L \text{ then} \\
&\text{\{} \\
&\text{\{} \\
&\text{\}}} \\
&\text{\} }
\end{align*}$

Based on the algorithm to identify the inverse map of hedge and properties studied in [8], we can establish the inverse map for $2 - \text{Mono} - \text{HA} = (X, \{c^+, c^-, c^0\}, V, M, P, \leq)$ with a note.
we can consider $h^{-1}(x) = W$ with $x \in H(c^+)$ we can consider $h^{-1}(x) = VPC_1^-$ the smallest value of $H(c^-)$; if $h^{-1}(x) = 1$ with $x \in H(c^-)$ we can consider $h^{-1}(x) = VVc_1^-$ the biggest value of $H(c^-)$; if $h^{-1}(x) = 0$ with $x \in H(c^-)$ we can consider $h^{-1}(x) = VVc_1^-$ 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>
<thead>
<tr>
<th>$V^-$</th>
<th>$M^-$</th>
<th>$P^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$kVc^-$</td>
<td>$Vc^-$</td>
<td>$VPC_1^-$</td>
</tr>
<tr>
<td>$sc^-$</td>
<td>$Vc_1^-$</td>
<td>$Vc_1^-$</td>
</tr>
<tr>
<td>$VPC_2^-$</td>
<td>$VMc_1^-$</td>
<td>$PMc_2^-$</td>
</tr>
<tr>
<td>$MPc_2^-$</td>
<td>$MMc_2^-$</td>
<td>$PC_2^-$</td>
</tr>
<tr>
<td>$Pc_2^-$</td>
<td>$Mc_2^-$</td>
<td>$PC_2^-$</td>
</tr>
<tr>
<td>$PPc_2^-$</td>
<td>$PMc_2^-$</td>
<td>$Vc_2^-$</td>
</tr>
<tr>
<td>$W$</td>
<td>$W$</td>
<td>$W$</td>
</tr>
<tr>
<td>$PPc_2^+$</td>
<td>$PPc_2^+$</td>
<td>$Vc_2^+$</td>
</tr>
<tr>
<td>$Pc_2^+$</td>
<td>$Pc_2^+$</td>
<td>$Vc_1^+$</td>
</tr>
<tr>
<td>$MPc_2^+$</td>
<td>$MPc_2^+$</td>
<td>$PPc_2^+$</td>
</tr>
<tr>
<td>$Vc_2^+$</td>
<td>$Vc_2^+$</td>
<td>$VMc_2^+$</td>
</tr>
<tr>
<td>$c^+$</td>
<td>$c^+$</td>
<td>$kMc_2^+$</td>
</tr>
<tr>
<td>$kVc_2^+$</td>
<td>$kMc_2^+$</td>
<td>$VVc_2^+$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 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 $AX = \{v_i, i = 1, 2, ..., n\}$ with $v_1 = 0$ and $v_n = 1$ in finite monotonic hedge algebra and linear order or $AX = \{v_i, i = 1, 2, ..., n; v_i = 0, v_n = 1 \ and \ 1 \leq i, j \leq n; v_i \geq v_j \Leftrightarrow i \geq j\}$. In linguistic many-valued logic, the truth degree of proposition is $v_i \in AX$.

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

$$(p(x; u), c_\delta) \Leftrightarrow (p(x; u), \delta h c) \ (RT1)$$

**Generalized Modus Tollens with Linguistic Modifiers**

$$(p(x; u), \delta c) \Leftrightarrow (p(x; hu), h(\delta c)) \ (GRT2)$$

With $h$ is a hedge and $\delta h c$, $h^{-1}(\delta c) \in AX$

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 defined by [9-11,15]:

$$T_L(v_i, v_j) = v_{\max(1,i+j-N)}$$

$$S_L(v_i, v_j) = v_{\min(N-i+j)}$$

$$I_L(v_i, v_j) = v_{\min(N,N-i+j)}$$

$\forall i, j \in AX$.

**Lemma 1** Let $v_i, v_j \in AX$, we have:

$$I_L(v_i, v_j) = I_L(N(v_j), N(v_i))$$

**Proof**

We have:

$$N(v_i) = v_{N-i+1}$$

$$N(v_j) = v_{N-j+1}$$

So,$$I_L(N(v_j), N(v_i)) = v_{\min(N,N-(N-j+1)+(N-i+1))}$$

$$= v_{\min(N,N-i+j)}$$

$$= I_L(v_i, v_j)$$

$\blacksquare$

### 4. Generalized modus ponens with linguistic modifiers

#### 4.1. Rule equivalent

In linguistic many-valued logic, implication satisfies the principle of the law of contrapositive symmetry (CPS) to obtain an equivalent rule from the given rule and investigate GMP for the execution of the rule in our inverse form of the approximate reasoning scheme.

Therefore, to develop inverse approximate reasoning, accoding to Lemma 1, we have the premise $a \rightarrow b$ can equivalently be transformed in to $\neg b \rightarrow \neg a$, that means:
“If X is A Then Y is B” may equivalently be transformed in to “If Y is NOT B Then X is NOT A” with NOT A and NOT B are the standard fuzzy complements of fuzzy sets A and B, respectively. Otherwise, the scheme of inverse approximate reasoning:

<table>
<thead>
<tr>
<th>Antecedent 1</th>
<th>If</th>
<th>X is A</th>
<th>Then</th>
<th>Y is B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent 2</td>
<td>Y is B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td>X is A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

may be rewritten as that in following scheme:

**Inverse Approximate Reasoning by GMP**

<table>
<thead>
<tr>
<th>Antecedent 1</th>
<th>If</th>
<th>Y is NOT B</th>
<th>Then</th>
<th>X is NOT A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent 2</td>
<td>Y is B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td>X is A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In [17], Banibrata Mondal and Swapan Raha investigated the inverse approximate reasoning based on GMP by fuzzy abduction to obtain A*. In our method, we proposed new generalized modus tollens with linguistic modifiers in many-valued logic for solving the inverse approximate reasoning scheme.

### 4.2. Generalized Modus Tollens with Linguistic Modifiers

One vague sentence can be represented by \( p(x;u) \), wherein \( x \) is a variable, \( u \) is a vague concept. In general, by an assertion is one pair \( A=(p(x;u), \delta c) \) (Symbol: \( (P,v) \)), wherein \( p(x;u) \) is a vague sentence, \( \delta c \) 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 derived rules. In [4-6], the hedge moving rules are set:

\[ RT1: \frac{(p(x;u),\delta c)}{(p(x;u),\delta c)} \]

\[ GRT2: \frac{(p(x;u),\delta c)}{(p(x;u),\delta c)} \]

And the following equivalent assertions:

\[ (p(x,\text{NOT}(u)), \delta c) \iff (p(x,u),\text{NOT}(\delta c)) \] (RN)

\[ (p(x,\text{NOT}(u)),\text{NOT}(u)) \iff (p(x,\delta(\text{NOT}(u)))) \] (RNH)

(with \( \delta \) is the hedge string)

In [9-10, 15], the generalized modus ponens was proposed.

\[ \frac{(p(x;u) \rightarrow q(y;v), p(x;u), v)}{(q(y;v), T_{L}(v,v))} \]

\[ \frac{(p(x;u), v)}{(q(y;v), T_{L}(v,v), v)} \]

Herein, EGMP is an extension of EGMP. From GMP, EGMP and RN, we have:

\[ NGMP: \frac{(p(x;\text{NOT}(u)), q(y,\text{NOT}(v)), v)}{(q(y;v), T_{L}(v,v), N(v,v))} \]

\[ NEGMP: \frac{(p(x;\text{NOT}(u)), v)}{(q(y;v), T_{L}(v,v), N(v,v))} \]

We can generalized modus tollens with linguistic modifiers as following:

Given \( \alpha, \beta, \delta, \theta, \delta', \alpha', \beta', \delta', \delta' \) are the hedge strings. We have the following propositions:

**Proposition 2**

\[
(p(x;\text{NOT}(\delta u)) \rightarrow q(y;\text{NOT}(\delta v)), \alpha c) \\
(q(y;\delta v), N(T_{L}(\alpha c, N(\delta^{-}(\alpha' \delta' \alpha' c))))
\]

**Proof**

According to RT1 we have: \( (p(x;u), \alpha' \delta' c) \); Then, applying GRT2 we have:

\[
(p(x;\delta u), \delta^{-}(-\alpha' \delta' c))
\]

Finally, using NGMP we have:

\[
(q(y;\delta v), N(T_{L}(\alpha c, N(\delta^{-}(\alpha' \delta' \alpha' c))))
\]

**Proposition 3**

\[
(p(x;\text{NOT}(\delta u)), \alpha c) \rightarrow (q(y;\text{NOT}(\delta v)), \beta c)
\]

\[
(p(x;\delta u), \alpha' \delta' c)
\]

\[
(q(y;\delta v), \delta^{-}(-T_{L}(I_{c}(N(\alpha c, \alpha c), N(\beta c), \delta' c)))
\]

**Proof**

Application RNH we have:

\[
p(x;\text{NOT}(\delta u)) \iff p(x;\delta(\text{NOT}(u))) \\
p(y;\text{NOT}(\delta v)) \iff p(y;\delta(\text{NOT}(v)))
\]

Applying RT1 we have:

\[
(p(x;\text{NOT}(u)), a \delta c); \quad (q(y;\text{NOT}(v)), \beta \delta c); \\
(p(x;u), \alpha' \delta' c)
\]

Then, using NEGMP we have:

\[
(q(y;\delta v), \delta^{-}(-T_{L}(I_{c}(N(\alpha c, \alpha c), N(\beta c), \alpha' \delta' c)))
\]

Finally, using GRT2 we have:

\[
(q(y;\delta v), \delta^{-}(-T_{L}(I_{c}(N(\alpha c, \alpha c), N(\beta c), \alpha' \delta' c)))
\]
Based on the above result, we have the new generalized modus tollens inference rules with linguistic modifiers by applying the following theorems:

**Theorem 1**

\[(q(y; δv) → p(x; δu)), ac\]

\[(p(x; δ'v), a'c)\]

\[(q(y; δv), N(T1p(ac, N(δ^-(a'δc)))))]

**Proof**

According to Lemma 1 and Proposition 2

**Theorem 2**

\[(q(y; δv), βc) → (p(x; δu), ac)\]

\[(p(x; δ'v), a'c)\]

\[(q(y; δ'v), δ^-(T1p(αδc, N(βδc)), a'δc))\]

**Proof**

According to Lemma 1 and Proposition 3

Based on Theorem 1 and Theorem 2, we build a deduction procedure for inverse approximate reasoning.

### 4.3. Deductive procedure based on generalized modus tollens with linguistic modifiers

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

Here, we build a deduction procedure (Algorithm 2) based on hedge moving rules and Theorem (2-3) for solving inverse approximate reasoning.

**Problem** Supposed 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 \( K; L \rightarrow Mono \rightarrow HA \)

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

**Method:**

1. **Step 1:** Using the moving rules RT1 and GRT2 to determine the dimension 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);

2. **Step 2:** Finding the truth value expression of the conclusion using Theorem (2-3);

3. **Step 3:** Transferring the truth value \( δc \) in the expression found in Step 2 into \( v1: v1 = δc \) (Algorithm 1);

4. **Step 4:** Calculating the truth value expression based on T-norms, T-conorm, implicators, negate operation was defined above an application inverse of hedge;

5. **Step 5:** Making the truth value of conclusion clause.

### 4.4. Examples

#### Example 6

Given the following knowledge base:

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

ii) Mary will be a more good employee is very possibly True

Find the truth value of the sentence: “Mary is studying more hard”

By formalizing: (i) – (ii), it can be rewritten as follows:

1. \((\text{studying(x; MHard)} → \text{emp(x; VGood)}, P \land True)\)

   (Based on the hypothesis)(i)

2. \((\text{emp(x; Good)}, P \land True)\)

   (Based on (ii))
Based on the knowledge base (i-ii) and Theorem 1, we have the following result:

\[(\text{studying}(\text{Mary}; \text{MHard}), N(T_L(\text{PVTrue}, N(\text{V}^{-}(\text{PVTrue}))))\]

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

\[N(\text{V}^{-}(\text{PVTrue})) = N(\text{MTrue}) = N(v_{12}) = v_{12}\]

\[\text{PVTrue} = v_{25}\]

\[N(T_L(\text{PVTrue}, N(\text{V}^{-}(\text{PVTrue})))) = N(T_L(v_{25}, v_{12})) = N(v_{18}) = v_{22} = \text{MTrue}\]

Therefore, the truth value of the sentence “Mary is studying possibly hard” is \((\text{studying}(\text{Mary}; \text{PHard}), \text{PVTrue}))\), which means Mary is studying possibly hard is **Possibly Very 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 linear order organized elements can act as base values set for truth domain of logic system. In this paper, we study new generalized modus tollens inference rules with linguistic modifiers build a deduction procedure and use it to solve the inverse approximate reasoning problem. In future researches, we would study an inference formalization for inverse approximate reasoning with more complex rules.

Acknowledgment

The authors would like to express deep gratitude to Vietnam’s National Foundation for Science and Technology Development (NAFOSTED) for its support and to those who reviewed the paper and gave comments for its completion.

References