INTERVAL NEUTROSOPHIC SETS AND LOGIC

THEORY AND APPLICATIONS IN COMPUTING

Prof. FLORENTIN SMARANDACHE, PhD

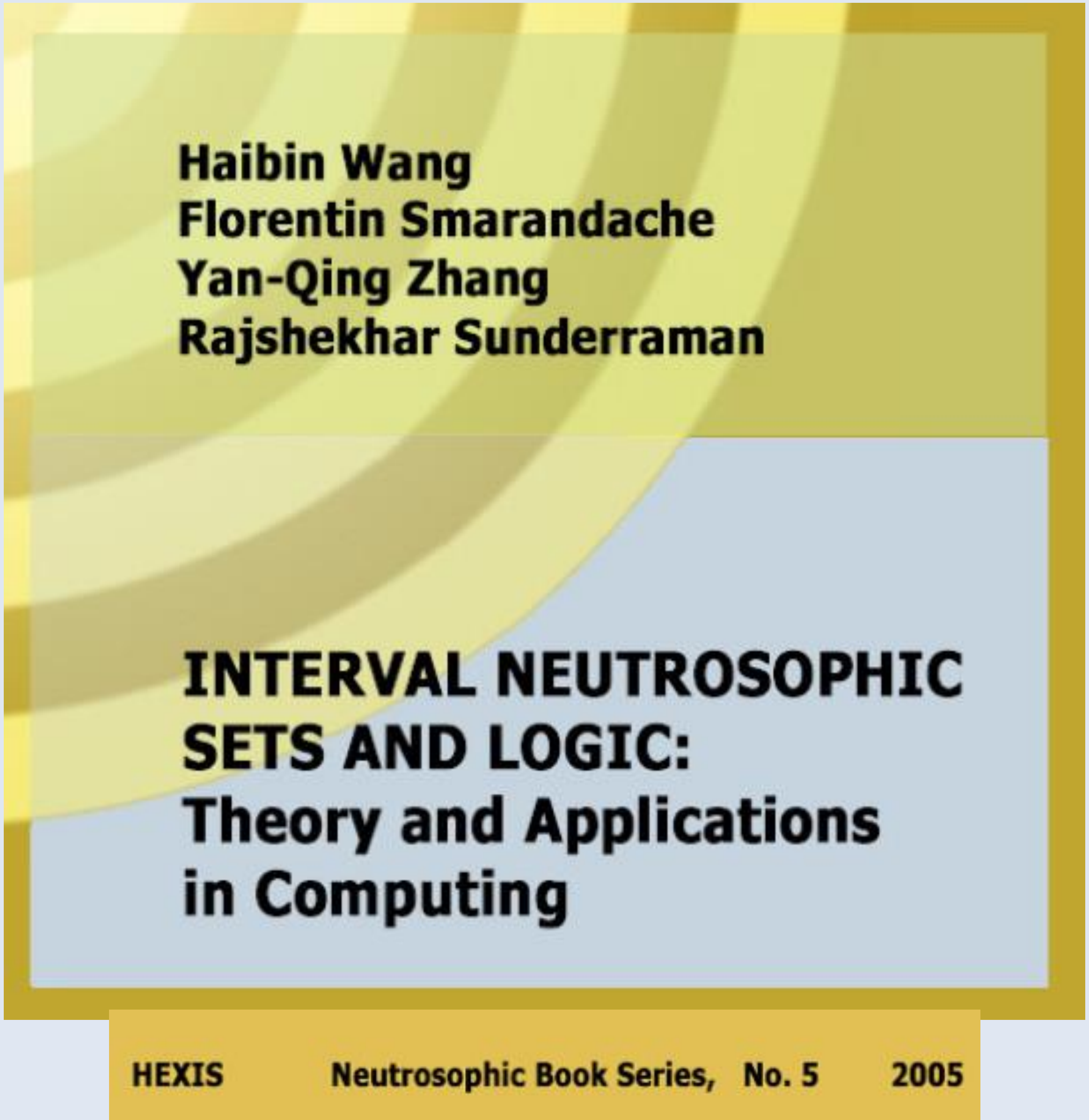
The University of New Mexico

Math & Science Dept.

705 Gurley Ave.

Gallup, NM 87301, USA

http://fs.gallup.unm.edu/


Haibin Wang
Florentin Smarandache
Yan-Qing Zhang
Rajshekhar Sunderraman
INTERVAL NEUTROSOPHIC SETS AND LOGIC:
Theory and Applications in Computing
HEXIS
Neutrosophic Book Series, No. 5
2005

Contents

1. Neutrosophy: Generalities

Neutrosophy is a new branch of philosophy that studies the origin, nature, and scope of neutralities, as well as their interactions with different ideational spectra.

The neutrosophics were introduced by F. Smarandache in 1995.

This theory considers every notion or idea <A> together with its opposite or negation <Anti-A> and the spectrum of "neutralities" <Neut-A> (i.e. notions or ideas located between the two extremes, supporting neither <A> nor <Anti-A>).

The <Neut-A> and <Anti-A> ideas together are referred to as <Non-A>.

According to this theory, every idea <A> tends to be neutralized and balanced by <Anti-A> and <Non-A> ideas - as a state of equilibrium.

In a classical way, <A>, <Neut-A>, <Anti-A> are disjoint two by two.

But, since in many cases the borders between notions are vague, imprecise, it is possible that <A>, <Neut-A>, <Anti-A> (and <Non-A>, of course) have common parts two by two as well.

Neutrosophy is the base of *neutrosophic logic*, *neutrosophic set*, *neutrosophic probability* and *neutrosophic statistics.*

Neutrosophic Logic is a general framework for unification of many existing logics.

The main idea of NL is to characterize each logical statement in a 3D Neutrosophic Space, where each dimension of the space represents respectively the truth (T), the falsehood (F), and the indeterminacy (I) of the statement under consideration, where T, I, F are standard or non-standard real subsets of $]0^-,1^+[$.

For software engineering proposals the classical unit interval [0,1] can be used.

T, I, F are independent components, leaving room for incomplete information (when their superior sum < 1), paraconsistent and contradictory information (when the superior sum > 1), or complete information (sum of components $= 1$).

As an example: a statement can be between [0.4,0.6] true, 0.1 or between (0.15,0.25) indeterminate, and either 0.4 or 0.6 false.

Neutrosophic Probability is a generalization of the classical probability and imprecise probability in which the chance that an event A occurs is $t\%$ true - where t varies in the subset T, $i\%$ indeterminate - where i varies in the subset I, and $f\%$ false - where f varies in the subset F.

In classical probability $nsup \leq 1$, while in neutrosophic probability $nsup \leq 3+$.

In imprecise probability: the probability of an event is a subset T in $[0,1]$, not a number p in $[0,1]$, what's left is supposed to be the opposite, subset F (also from the unit interval $[0,1]$); there is no indeterminate subset I in imprecise probability.

Neutrosophic Statistics is the analysis of events described by the neutrosophic probability.

The function that models the neutrosophic probability of a random variable x is called neutrosophic distribution:

$$NP(x) = (T(x), I(x), F(x)),$$

where $T(x)$ represents the probability that value x occurs, $F(x)$ represents the probability that value x does not occur, and $I(x)$ represents the indeterminate / unknown probability of value x.

Neutrosophic Set

Let *U* be a universe of discourse, and *M* a set included in *U*.

An element *x* from *U* is noted with respect to the set *M* as *x(T, I, F)* and belongs to *M* in the following way:

- *t*% true in the set,
- *i*% indeterminate (unknown if it is) in the set, and
- *f*% false,

where *t* varies in *T*, *i* varies in *I*, *f* varies in *F*.

Statically *T*, *I*, *F* are subsets, but dynamically *T*, *I*, *F* are functions/operators depending on many known or unknown parameters.

Neutrosophic set is a powerful general formal framework which generalizes the concept of the classic set, fuzzy set, interval valued fuzzy set, intuitionistic fuzzy set, interval valued intuitionistic fuzzy set, paraconsistent set, dialetheist set, paradoxist set, tautological set.

Neutrosophy World is expanding with **new research subjects**:

1. Neutrosophic topology including neutrosophic metric spaces and smooth topological spaces
2. Neutrosophic numbers and arithmetical operations, including ranking procedures for neutrosophic numbers
3. Neutrosophic rough sets
4. Neutrosophic relational structures, including neutrosophic relational equations, neutrosophic similarity relations, and neutrosophic orderings,
5. Neutrosophic geometry
6. Neutrosophic probability
7. Neutrosophic logical operations, including n-norms, n-conorms, neutrosophic implicators, neutrosophic quantifiers
8. Measures of neutrosophication
9. Deneutrosophication techniques
10. Neutrosophic measures and neutrosophic integrals

11. Neutrosophic multivalued mappings
12. Neutrosophic differential calculus
13. Neutrosophic mathematical morphology
14. Neutrosophic algebraic structures
15. Neutrosophic models
16. Neutrosophic cognitive maps
17. Neutrosophic matrix
18. Neutrosophic graph
19. Neutrosophic fusion rules
20. Neutrosophic relational maps
21. Applications:

    neutrosophic relational databases, neutrosophic image processing, neutrosophic linguistic variables, neutrosophic decision making and preference structures, neutrosophic expert systems, neutrosophic reliability theory, neutrosophic soft computing techniques in e-commerce and e-learning

2. Interval Neutrosophic Sets

Introduction

The neutrosophic set generalizes the classic set, the fuzzy set, the interval valued fuzzy set, the intuitionistic fuzzy set, the interval valued intuitionistic fuzzy set, the paraconsistent set, the dialetheist set, the paradoxist set, the tautological set - from philosophical point of view.

From scientific or engineering point of view, the neutrosophic set and set-theoretic operators need to be specified.

Otherwise, it will be difficult to apply in the real applications.

We define the set-theoretic operators on an instance of neutrosophic set called Interval Neutrosophic Set (INS).

We call it "interval" because it is a subclass of neutrosophic set, that is we only consider the subunitary interval of $[0,1]$.

An interval neutrosophic set A is defined on universe X, $x = x(T,I,F) \in A$ with T, I and F being the subinterval of $[0,1]$.

The interval neutrosophic set can represent uncertain, imprecise, incomplete and inconsistent information which exist in real world.

The interval neutrosophic set generalizes the following sets:

1. the *classical set*, $I = \emptyset$, $\inf T = \sup T = 0$ or 1, $\inf F = \sup F = 0$ or 1 and $\sup T + \sup F = 1$.
2. the *fuzzy set*, $I = \emptyset$, $\inf T = \sup T \in [0,1]$, $\inf F = \sup F \in [0,1]$ and $\sup T + \sup F = 1$.
3. the *interval valued fuzzy set*, $I = \emptyset$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0,1]$,
$\sup T + \inf F = 1$ and $\inf T + \sup F = 1$.
4. the *intuitionistic fuzzy set*, $I = \emptyset$, $\inf T = \sup T \in [0,1]$, $\inf F = \sup F \in [0,1]$ and $\sup T + \sup F \leq 1$.
5. the *interval valued intuitionistic fuzzy set*, $I = \emptyset$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0,1]$ and $\sup T + \sup F \leq 1$.
6. the *paraconsistent set*, $I = \emptyset$, $\inf T = \sup T \in [0,1]$, $\inf F = \sup F \in [0,1]$ and $\sup T + \sup F > 1$.
7. the *interval valued paraconsistent set*, $I = \emptyset$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0,1]$ and $\inf T + \inf F > 1$.

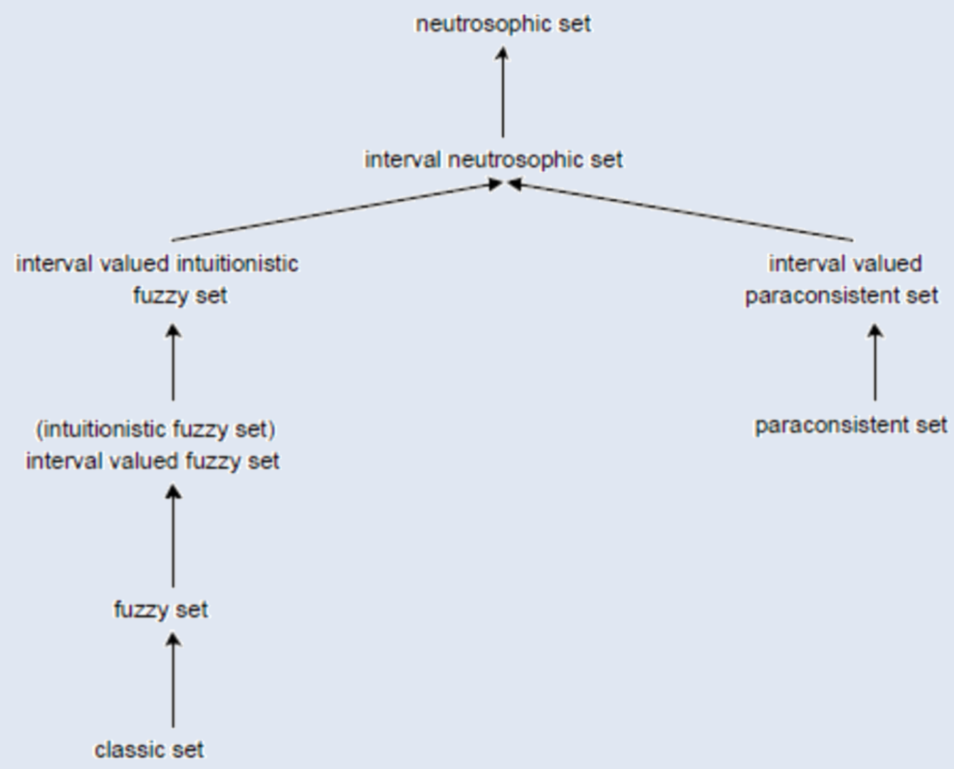


Figure 1: Relationship among interval neutrosophic set and other sets

Note: →, such as a → b, means that b is a generalization of a.

Neutrosophic Set

Definition 1

Let X be a space of points (objects), with a generic element in X denoted by x.

A neutrosophic set A in X is characterized by a truth-membership function T_A, a indeterminacy-membership function I_A and a falsity-membership function F_A. $T_A(x)$, $I_A(x)$ and $F_A(x)$ are real standard or non-standard subsets of $]0^-,1^+[$.

That is

$$T_A : X \to]0^-,1^+[, \quad (1.1)$$

$$I_A : X \to]0^-,1^+[, \quad (1.2)$$

$$F_A : X \to]0^-,1^+[. \quad (1.3)$$

There is no restriction on the sum of $T_A(x)$, $I_A(x)$ and $F_A(x)$, so $0^- \leq \sup T_A(x) + \sup I_A(x) + \sup F_A(x) \leq 3^+$.

Definition 2

The complement of a neutrosophic set A is denoted by $\bar{A}$ and is defined by

$$T_{\bar{A}}(x) = \{1^+\} \ominus T_A(x), \tag{1.4}$$
$$I_{\bar{A}}(x) = \{1^+\} \ominus I_A(x), \tag{1.5}$$
$$F_{\bar{A}}(x) = \{1^+\} \ominus F_A(x), \tag{1.6}$$

for all x in X.

Definition 3 (Containment)

A neutrosophic set A is contained in the other neutrosophic set B, $A \subseteq B$, if and only if

$$\inf T_A(x) \le \inf T_B(x) \quad , \quad \sup T_A(x) \le \sup T_B(x), \tag{1.7}$$

$$\inf I_A(x) \ge \inf I_B(x) \quad , \quad \sup I_A(x) \ge \sup I_B(x), \tag{1.8}$$

$$\inf F_A(x) \ge \inf F_B(x) \quad , \quad \sup F_A(x) \ge \sup F_B(x), \tag{1.9}$$

for all x in X.

Definition 4 (Union)

The union of two neutrosophic sets A and B is a neutrosophic set C, written as $C = A \cup B$, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$
\begin{aligned}
T_C(x) &= T_A(x) \oplus T_B(x) \ominus T_A(x) \odot T_B(x), && (1.10)\\
I_C(x) &= I_A(x) \oplus I_B(x) \ominus I_A(x) \odot I_B(x), && (1.11)\\
F_C(x) &= F_A(x) \oplus F_B(x) \ominus F_A(x) \odot F_B(x), && (1.12)
\end{aligned}
$$

for all x in X.

Definition 5 (Intersection)

The intersection of two neutrosophic sets A and B is a neutrosophic set C, written as $C = A \cap B$, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$T_C(x) = T_A(x) \odot T_B(x), \quad (1.13)$$

$$I_C(x) = I_A(x) \odot I_B(x), \quad (1.14)$$

$$F_C(x) = F_A(x) \odot F_B(x), \quad (1.15)$$

for all x in X.

Definition 6 (Difference)

The difference of two neutrosophic sets A and B is a neutrosophic set C, written as $C = A \setminus B$, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$T_C(x) = T_A(x) \ominus T_A(x) \odot T_B(x), \tag{1.16}$$
$$I_C(x) = I_A(x) \ominus I_A(x) \odot I_B(x), \tag{1.17}$$
$$F_C(x) = F_A(x) \ominus F_A(x) \odot F_B(x), \tag{1.18}$$

for all x in X.

Definition 7 (Cartesian Product)

Let A be the neutrosophic set defined on universe E_1 and B be the neutrosophic set defined on universe E_2.

If

$$x(T_A^1, I_A^1, F_A^1) \in A \ and \ y(T_A^2, I_A^2, F_A^2) \in B,$$

then the cartesian product of two neutrosophic sets A and B is defined by

$$(x(T_A^1, I_A^1, F_A^1), y(T_A^2, I_A^2, F_A^2)) \in A \times B \tag{1.19}$$

Interval Neutrosophic Set

The interval neutrosophic set (INS) is an instance of neutrosophic set which can be used in real scientific and engineering applications.

Definition 8 (Interval Neutrosophic Set)

Let X be a space of points (objects), with a generic element in X denoted by x. An interval neutrosophic set (INS) A in X is characterized by truth-membership function T_A, indeterminacy membership function I_A and falsity-membership function F_A. For each point x in X, $T_A(x)$, $I_A(x)$, $F_A(x) \subseteq [0,1]$.

An interval neutrosophic set (INS) in R^1 is illustrated in the figure below.

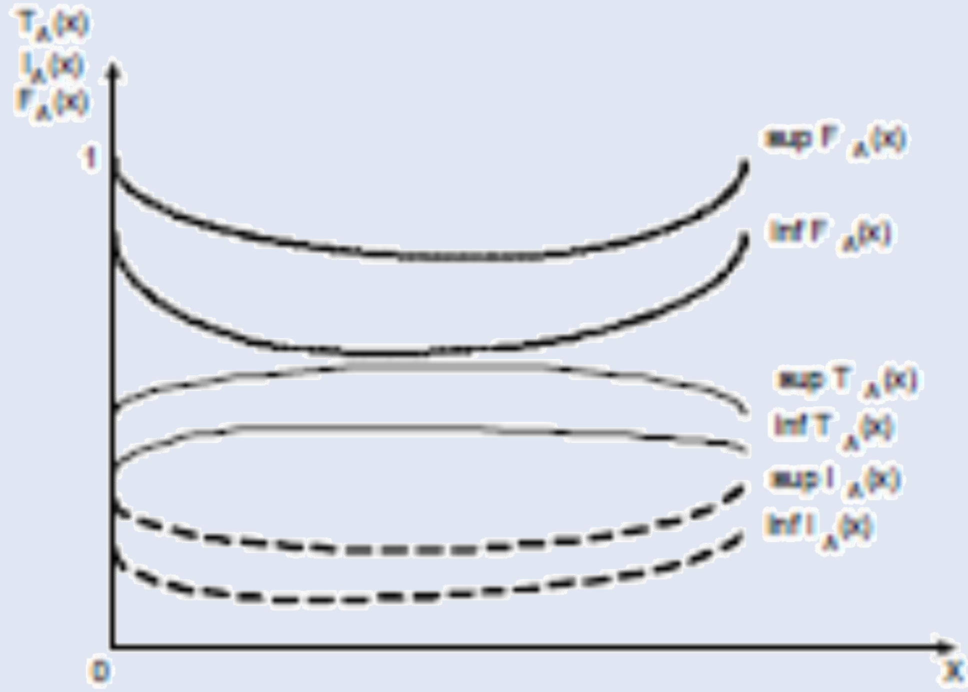


Figure 2: Illustration of interval neutrosophic set in R^1

When X is continuous, an INS A can be written as

$$A = \int_X \langle T(x), I(x), F(x) \rangle / x, \;\; x \in X \tag{1.20}$$

When X is discrete, an INS A can be written as

$$A = \sum_{i=1}^{n} \langle T(x_i), I(x_i), F(x_i) \rangle / x_i, \;\; x_i \in X \tag{1.21}$$

Consider parameters such as capability, trustworthiness and price of semantic Web services.

These parameters are commonly used to define quality of service of semantic Web services.

We will use the evaluation of quality of service of semantic Web services as running example to illustrate every set-theoretic operation on interval neutrosophic set.

H. B. Wang, Y. Q. Zhang, and R. Sunderraman, *Soft semantic web services agent*, The Proceedings of NAFIPS 2004, 2004, pp. 126–129.

Example 1

Assume that $X = [x_1, x_2, x_3]$. x_1 is capability, x_2 is trustworthiness and x_3 is price. The values of x_1, x_2 and x_3 are in [0,1]. They are obtained from the questionnaire of some domain experts, their option could be degree of good, degree of indeterminacy and degree of poor. A is an interval neutrosophic set of X defined by

$$A = \langle [0.2, 0.4], [0.3, 0.5], [0.3, 0.5] \rangle / x_1 + \langle [0.5, 0.7], [0, 0.2], [0.2, 0.3] \rangle / x_2 + \langle [0.6, 0.8], [0.2, 0.3], [0.2, 0.3] \rangle / x_3 .$$

B is an interval neutrosophic set of X defined by

$$B = \langle [0.5, 0.7], [0.1, 0.3], [0.1, 0.3] \rangle / x_1 + \langle [0.2, 0.3], [0.2, 0.4], [0.5, 0.8] \rangle / x_2 + \langle [0.4, 0.6], [0, 0.1], [0.3, 0.4] \rangle / x_3 .$$

Definition 9

An interval neutrosophic set A is empty if and only if its

$$\begin{aligned} \inf T_A(x) &= \sup T_A(x) = 0, \\ \inf I_A(x) &= \sup I_A(x) = 1 \\ \text{and } \inf F_A(x) &= \sup T_A(x) = 0, \end{aligned}$$

for all x in X.

We now present the set-theoretic operators on interval neutrosophic set.

Definition 10 (Containment)

An interval neutrosophic set A is contained in the other interval neutrosophic set B, $A \subseteq B$, if and only if

$$\inf T_A(x) \leq \inf T_B(x) \quad , \quad \sup T_A(x) \leq \sup T_B(x), \tag{1.22}$$

$$\inf I_A(x) \geq \inf I_B(x) \quad , \quad \sup I_A(x) \geq \sup I_B(x), \tag{1.23}$$

$$\inf F_A(x) \geq \inf F_B(x) \quad , \quad \sup F_A(x) \geq \sup F_B(x), \tag{1.24}$$

for all x in X.

Definition 11

Two interval neutrosophic sets A and B are equal, written as $A = B$, if and only if $A \subseteq B$ and $B \subseteq A$.

Let $\underline{0} = \langle 0, 1, 1 \rangle$ and $\underline{1} = \langle 1, 0, 0 \rangle$.

Definition 12 (Complement)

Let C_N denote a neutrosophic complement of A. Then C_N is a function

$$C_N : N \longrightarrow N$$

and C_N must satisfy at least the following three axiomatic requirements:

1. $CN(\underline{0}) = 1$ and $CN(\underline{1}) = \underline{0}$

(boundary conditions)

2. Let A and B be two interval neutrosophic sets defined on X, if $A(x) \leq B(x)$, then $CN(A(x)) \geq CN(B(x))$, for all x in X

(monotonicity)

3. Let A be an interval neutrosophic set defined on X, then $CN(CN(A(x))) = A(x)$, for all x in X

(involutivity)

Definition 13 (Complement C_{N_1})

The complement of an interval neutrosophic set A is denoted by $\bar{A}$ and is defined by

$$
\begin{aligned}
T_{\bar{A}}(x) &= F_A(x), && (1.25)\\
\inf I_{\bar{A}}(x) &= 1 - \sup I_A(x), && (1.26)\\
\sup I_{\bar{A}}(x) &= 1 - \inf I_A(x), && (1.27)\\
F_{\bar{A}}(x) &= T_A(x), && (1.28)
\end{aligned}
$$

for all x in X.

Example 2

Let A be the interval neutrosophic set defined in Example 1. Then,

$$\bar{A} = \langle [0.3, 0.5], [0.5, 0.7], [0.2, 0.4] \rangle / x_1 + \langle [0.2, 0.3], [0.8, 1.0], [0.5, 0.7] \rangle / x_2 + \langle [0.2, 0.3], [0.7, 0.8], [0.6, 0.8] \rangle / x_3 .$$

Definition 14 (N-norm)

Let I_N denote a neutrosophic intersection of two interval neutrosophic sets A and B. Then I_N is a function

$$I_N : N \times N \rightarrow N$$

and I_N must satisfy at least the following four axiomatic requirements:

1. $I_N(A(x), \underline{1}) = A(x)$, for all x in X.

   (boundary conditions)

2. $B(x) \leq C(x)$ implies $I_N(A(x), B(x)) \leq I_N(A(x), C(x))$, for all x in X.

   (monotonicity)

3. $I_N(A(x), B(x)) = I_N(B(x), A(x))$, for all x in X.

   (commutativity)

4. $I_N(A(x), I_N(B(x), C(x))) = I_N(I_N(A(x), B(x)), C(x))$, for all x in X.

   (associativity)

Definition 15 (Intersection I_{N_1})

The intersection of two interval neutrosophic sets A and B is an interval neutrosophic set C, written as $C = A \cap B$, whose truth-membership, indeterminacy-membership, and false-membership are related to those of A and B by

$$\begin{aligned}
\inf T_C(x) &= \min(\inf T_A(x), \inf T_B(x)), && (1.29)\\
\sup T_C(x) &= \min(\sup T_A(x), \sup T_B(x)), && (1.30)\\
\inf I_C(x) &= \max(\inf I_A(x), \inf I_B(x)), && (1.31)\\
\sup I_C(x) &= \max(\sup I_A(x), \sup I_B(x)), && (1.32)\\
\inf F_C(x) &= \max(\inf F_A(x), \inf F_B(x)), && (1.33)\\
\sup F_C(x) &= \max(\sup F_A(x), \sup F_B(x)), && (1.34)
\end{aligned}$$

for all x in X.

Example 3

Let A and B be the interval neutrosophic sets defined in Example 1.

Then,

$$A \cap B = \begin{array}{l} \langle [0.2, 0.4], [0.3, 0.5], [0.3, 0.5] \rangle / x_1 + \\ \langle [0.2, 0.3], [0.2, 0.4], [0.5, 0.8] \rangle / x_2 + \langle [0.4, 0.6], [0.2, 0.3], [0.3, 0.4] \rangle / x_3 . \end{array}$$

Theorem 1

$A \cap B$ is the largest interval neutrosophic set contained in both A and B.

Definition 16 (N-conorm)

Let U_N denote a neutrosophic union of two interval neutrosophic sets A and B. Then U_N is a function

$$U_N : N \times N \rightarrow N$$

and U_N must satisfy at least the following four axiomatic requirements:

1. $U_N(A(x), \underline{0}) = A(x)$, *for all* x *in* X. (boundary conditions)

2. $B(x) \leq C(x)$ *implies* $U_N(A(x), B(x)) \leq U_N(A(x), C(x))$, *for all* x *in* X. (monotonicity)

3. $U_N(A(x), B(x)) = U_N(B(x), A(x))$, *for all* x *in* X. (commutativity)

4. $U_N(A(x), U_N(B(x), C(x))) = U_N(U_N(A(x), B(x)), C(x))$, *for all* x *in* X. (associativity)

Definition 17 (Union U_{N_1})

The union of two interval neutrosophic sets A and B is an interval neutrosophic set C, written as $C = A \cup B$, whose truth-membership, indeterminacy-membership, and false membership are related to those of A and B by

$$\inf T_C(x) = \max(\inf T_A(x), \inf T_B(x)), \quad (1.35)$$
$$\sup T_C(x) = \max(\sup T_A(x), \sup T_B(x)), \quad (1.36)$$
$$\inf I_C(x) = \min(\inf I_A(x), \inf I_B(x)), \quad (1.37)$$
$$\sup I_C(x) = \min(\sup I_A(x), \sup I_B(x)), \quad (1.38)$$
$$\inf F_C(x) = \min(\inf F_A(x), \inf F_B(x)), \quad (1.39)$$
$$\sup F_C(x) = \min(\sup F_A(x), \sup F_B(x)), \quad (1.40)$$

for all x in X.

Example 4

Let A and B be the interval neutrosophic sets defined in Example 1.

Then,

$$A \cup B = \begin{array}{l} \langle [0.5, 0.7], [0.1, 0.3], [0.1, 0.3] \rangle / x_1 + \\ \langle [0.5, 0.7], [0, 0.2], [0.2, 0.3] \rangle / x_2 + \langle [0.6, 0.8], [0, 0.1], [0.2, 0.3] \rangle / x_3 . \end{array}$$

The intuition behind the union operator is that if one of elements in A and B is true then it is true in $A \cup B$, only both are indeterminate and false in A and B then it is indeterminate and false in $A \cup B$. The other operators should be understood similarly.

Theorem 2

$A \cup B$ is the smallest interval neutrosophic set containing both A and B.

Theorem 3

Let P be the power set of all interval neutrosophic sets defined in the universe X.

Then $\langle P; I_{N_1}, U_{N_1} \rangle$ is a distributive lattice.

Definition 18 (Interval neutrosophic relation)

Let X and Y be two non-empty crisp sets. An interval neutrosophic relation $R(X,Y)$ is a subset of product space $X \times Y$, and is characterized by the truth membership function $T_R(x,y)$, the indeterminacy membership function $I_R(x,y)$, and the falsity membership function $F_R(x,y)$, where $x \in X$ and $y \in Y$ and $T_R(x,y)$, $I_R(x,y)$, $F_R(x,y) \subseteq [0,1]$.

Definition 19 (Interval Neutrosophic Composition Functions)

The membership functions for the composition of interval neutrosophic relations *R(X,Y)* and *S(Y,Z)* are given by the interval neutrosophic sup-star composition of *R* and *S*

$$T_{R\circ S}(x,z) = \sup_{y\in Y} \min(T_R(x,y), T_S(y,z)), \quad (1.41)$$

$$I_{R\circ S}(x,z) = \sup_{y\in Y} \min(I_R(x,y), I_S(y,z)), \quad (1.42)$$

$$F_{R\circ S}(x,z) = \inf_{y\in Y} \max(F_R(x,y), F_S(y,z)). \quad (1.43)$$

If R is an interval neutrosophic set rather than an interval neutrosophic relation, then $Y = X$ and $\sup_{y\in Y} \min(T_R(x,y), T_S(y,z))$ becomes $\sup_{y\in Y} \min(T_R(x), T_S(y,z))$, which is only a function of the output variable z. It is similar for $\sup_{y\in Y} \min(I_R(x,y), I_S(y,z))$ and $\inf_{y\in Y} \max(F_R(x,y), F_S(y,z))$. Hence, the notation of $T_{R\circ S}(x,z)$ can be simplified to $T_{R\circ S}(z)$, so that in the case of R being just an interval neutrosophic set,

$$T_{R\circ S}(z) = \sup_{x\in X} \min(T_R(x), T_S(x,z)), \quad (1.44)$$

$$I_{R\circ S}(z) = \sup_{x\in X} \min(I_R(x), I_S(x,z)), \quad (1.45)$$

$$F_{R\circ S}(z) = \inf_{x\in X} \max(F_R(x), F_S(x,z)). \quad (1.46)$$

Definition 20 (Difference)

The difference of two interval neutrosophic sets A and B is an interval neutrosophic set C, written as $C = A \setminus B$, whose truth-membership, indeterminacy-membership and falsity membership functions are related to those of A and B by

$$\begin{aligned}
\inf T_C(x) &= \min(\inf T_A(x), \inf F_B(x)), && (1.47)\\
\sup T_C(x) &= \min(\sup T_A(x), \sup F_B(x)), && (1.48)\\
\inf I_C(x) &= \max(\inf I_A(x), 1 - \sup I_B(x)), && (1.49)\\
\sup I_C(x) &= \max(\sup I_A(x), 1 - \inf I_B(x)), && (1.50)\\
\inf F_C(x) &= \max(\inf F_A(x), \inf T_B(x)), && (1.51)\\
\sup F_C(x) &= \max(\sup F_A(x), \sup T_B(x)), && (1.52)
\end{aligned}$$

for all x in X.

Example 5

Let A and B be the interval neutrosophic sets defined in Example 1.

Then,

$$A \setminus B = \begin{array}{l}\langle[0.1, 0.3], [0.7, 0.9], [0.5, 0.7]\rangle/x_1 + \\ \langle[0.5, 0.7], [0.6, 0.8], [0.2, 0.3]\rangle/x_2 + \langle[0.3, 0.4], [0.9, 1.0], [0.4, 0.6]\rangle/x_3 .\end{array}$$

Theorem 4 $A \subseteq B \leftrightarrow \bar{B} \subseteq \bar{A}$

Definition 21 (Addition)

The addition of two interval neutrosophic sets A and B is an interval neutrosophic set C, written as $C = A + B$, whose truth-membership, indeterminacy-membership and falsity membership functions are related to those of A and B by

$$\begin{aligned}
\inf T_C(x) &= \min(\inf T_A(x) + \inf T_B(x), 1), && (1.53)\\
\sup T_C(x) &= \min(\sup T_A(x) + \sup T_B(x), 1), && (1.54)\\
\inf I_C(x) &= \min(\inf I_A(x) + \inf I_B(x), 1), && (1.55)\\
\sup I_C(x) &= \min(\sup I_A(x) + \sup I_B(x), 1), && (1.56)\\
\inf F_C(x) &= \min(\inf F_A(x) + \inf F_B(x), 1), && (1.57)\\
\sup F_C(x) &= \min(\sup F_A(x) + \sup F_B(x), 1), && (1.58)
\end{aligned}$$

for all x in X.

Example 6

Let A and B be the interval neutrosophic sets defined in Example 1. Then,

$$A + B = \begin{array}{l} \langle [0.7, 1.0], [0.4, 0.8], [0.4, 0.8] \rangle / x_1 + \\ \langle [0.7, 1.0], [0.2, 0.6], [0.7, 1.0] \rangle / x_2 + \langle [1.0, 1.0], [0.2, 0.4], [0.5, 0.7] \rangle / x_3 . \end{array}$$

Definition 22 (Cartesian product)

The cartesian product of two interval neutrosophic sets A defined on universe X_1 and B defined on universe X_2 is an interval neutrosophic set C, written as $C = A \times B$, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A and B by

$$\begin{aligned}
\inf T_C(x,y) &= \inf T_A(x) + \inf T_B(y) - \inf T_A(x) \cdot \inf T_B(y), && (1.59)\\
\sup T_C(x,y) &= \sup T_A(x) + \sup T_B(y) - \sup T_A(x) \cdot \sup T_B(y), && (1.60)\\
\inf I_C(x,y) &= \inf I_A(x) \cdot \sup I_B(y), && (1.61)\\
\sup I_C(x,y) &= \sup I_A(x) \cdot \sup I_B(y), && (1.62)\\
\inf F_C(x,y) &= \inf F_A(x) \cdot \inf F_B(y), && (1.63)\\
\sup F_C(x,y) &= \sup F_A(x) \cdot \sup F_B(y), && (1.64)
\end{aligned}$$

for all x in X_1 and y in X_2.

Example 7

Let A and B be the interval neutrosophic sets defined in Example 1. Then,

$$A \times B = \begin{array}{l} \langle [0.6, 0.82], [0.03, 0.15], [0.03, 0.15] \rangle / x_1 + \\ \langle [0.6, 0.79], [0, 0.08], [0.1, 0.24] \rangle / x_2 + \langle [0.76, 0.92], [0, 0.03], [0.03, 0.12] \rangle / x_3 . \end{array}$$

Definition 23 (Scalar multiplication)

The scalar multiplication of interval neutrosophic set A is an interval neutrosophic set B, written as $B = a \cdot A$, whose truth-membership, indeterminacy-membership and falsity-membership functions are related to those of A by

$$\inf T_B(x) = \min(\inf T_A(x) \cdot a, 1), \quad (1.65)$$
$$\sup T_B(x) = \min(\sup T_A(x) \cdot a, 1), \quad (1.66)$$
$$\inf I_B(x) = \min(\inf I_A(x) \cdot a, 1), \quad (1.67)$$
$$\sup I_B(x) = \min(\sup I_A(x) \cdot a, 1), \quad (1.68)$$
$$\inf F_B(x) = \min(\inf F_A(x) \cdot a, 1), \quad (1.69)$$
$$\sup F_B(x) = \min(\sup F_A(x) \cdot a, 1), \quad (1.70)$$

for all x in X and $a \in R^+$.

Definition 24 (Scalar division)

The scalar division of interval neutrosophic set A is an interval neutrosophic set B, written as $B = a \cdot A$, whose truth-membership, indeterminacy-membership and falsity membership functions are related to those of A by

$$\inf T_B(x) = \min(\inf T_A(x)/a, 1), \quad (1.71)$$
$$\sup T_B(x) = \min(\sup T_A(x)/a, 1), \quad (1.72)$$
$$\inf I_B(x) = \min(\inf I_A(x)/a, 1), \quad (1.73)$$
$$\sup I_B(x) = \min(\sup I_A(x)/a, 1), \quad (1.74)$$
$$\inf F_B(x) = \min(\inf F_A(x)/a, 1), \quad (1.75)$$
$$\sup F_B(x) = \min(\sup F_A(x)/a, 1), \quad (1.76)$$

for all x in X and $a \in R^+$.

Now we will define two operators: truth-favorite (Δ) and false-favorite (∇) to remove the indeterminacy in the interval neutrosophic sets and transform it into interval valued intuitionistic fuzzy sets or interval valued paraconsistent sets. These two operators are unique on interval neutrosophic sets.

Definition 25 (Truth-favorite)

The truth-favorite of interval neutrosophic set A is an interval neutrosophic set B, written as $B = \Delta A$, whose truth-membership and falsity-membership functions are related to those of A by

$$
\begin{aligned}
\inf T_B(x) &= \min(\inf T_A(x) + \inf I_A(x), 1), && (1.77)\\
\sup T_B(x) &= \min(\sup T_A(x) + \sup I_A(x), 1), && (1.78)\\
\inf I_B(x) &= 0, && (1.79)\\
\sup I_B(x) &= 0, && (1.80)\\
\inf F_B(x) &= \inf F_A(x), && (1.81)\\
\sup F_B(x) &= \sup F_A(x), && (1.82)
\end{aligned}
$$

for all x in X.

Definition 26 (False-favorite)

The False-favorite of interval neutrosophic set A is an interval neutrosophic set B, written as $B = \nabla A$, whose truth-membership and falsity-membership functions are related to those of A by

$$\begin{aligned}
\inf T_B(x) &= \inf T_A(x), && (1.83)\\
\sup T_B(x) &= \sup T_A(x), && (1.84)\\
\inf I_B(x) &= 0, && (1.85)\\
\sup I_B(x) &= 0, && (1.86)\\
\inf F_B(x) &= \min(\inf F_A(x) + \inf I_A(x), 1), && (1.87)\\
\sup F_B(x) &= \min(\sup F_A(x) + \sup I_A(x), 1), && (1.88)
\end{aligned}$$

for all x in X.

Example 8

Let A and B be the interval neutrosophic sets defined in Example 1. Then,

$$\triangle A = \langle [0.5, 0.9], [0, 0], [0.3, 0.5] \rangle / x_1 + \langle [0.5, 0.9], [0, 0], [0.2, 0.3] \rangle / x_2 + \langle [0.8, 1.0], [0, 0], [0.2, 0.3] \rangle / x_3 .$$

The purpose of truth-favorite operator is to evaluate the maximum of degree of truth-membership of interval neutrosophic set.

Example 9

Let A and B be the interval neutrosophic sets defined in Example 1. Then,

$$\nabla A = \langle [0.2, 0.4], [0, 0], [0.6, 1.0] \rangle / x_1 + \langle [0.5, 0.7], [0, 0], [0.2, 0.5] \rangle / x_2 + \langle [0.6, 0.8], [0, 0], [0.4, 0.6] \rangle / x_3 .$$

The purpose of false-favorite operator is to evaluate the maximum of degree of false-membership of interval neutrosophic set.

Theorem 5

For every two interval neutrosophic sets A and B:

1. $\triangle(A \cup B) \subseteq \triangle A \cup \triangle B$
2. $\triangle A \cap \triangle B \subseteq \triangle(A \cap B)$
3. $\nabla A \cup \nabla B \subseteq \nabla(A \cup B)$
4. $\nabla(A \cap B) \subseteq \nabla A \cap \nabla B$

Properties of Set-theoretic Operators

Property 1 (Commutativity) $A \cup B = B \cup A$, $A \cap B = B \cap A$, $A + B = B + A$, $A \times B = B \times A$

Property 2 (Associativity) $A \cup (B \cup C) = (A \cup B) \cup C$,
$A \cap (B \cap C) = (A \cap B) \cap C$,
$A + (B + C) = (A + B) + C$,
$A \times (B \times C) = (A \times B) \times C$.

Property 3 (Distributivity) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Property 4 (Idempotency) $A \cup A = A$, $A \cap A = A$, $\triangle\triangle A = \triangle A$, $\nabla\nabla A = \nabla A$.

Property 5 $A \cap \Phi = \Phi$, $A \cup X = X$, *where* $\inf T_\Phi = \sup T_\Phi = 0$, $\inf I_\Phi = \sup I_\Phi = \inf F_\Phi = \sup F_\Phi = 1$ *and* $\inf T_X = \sup T_X = 1$, $\inf I_X = \sup I_X = \inf F_X = \sup F_X = 0$.

Property 6 $\triangle(A + B) = \triangle A + \triangle B$, $\nabla(A + B) = \nabla A + \nabla B$.

Property 7 $A \cup \Psi = A$, $A \cap X = A$, *where* $\inf T_\Phi = \sup T_\Phi = 0$, $\inf I_\Phi = \sup I_\Phi = \inf F_\Phi = \sup F_\Phi = 1$ *and* $\inf T_X = \sup T_X = 1$, $\inf I_X = \sup I_X = \inf F_X = \sup F_X = 0$.

Property 8 (Absorption) $A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$

Property 9 (DeMorgan's Laws) $\overline{A \cup B} = \bar{A} \cap \bar{B}$, $\overline{A \cap B} = \bar{A} \cup \bar{B}$.

Property 10 (Involution) $\overline{\overline{A}} = A$

Convexity of Interval Neutrosophic Set

We assume that X is a real Euclidean space E^n for correctness.

Definition 27 (Convexity)

An interval neutrosophic set A is convex if and only if

$$\inf T_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\inf T_A(x_1), \inf T_A(x_2)), \quad (1.89)$$
$$\sup T_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\sup T_A(x_1), \sup T_A(x_2)), \quad (1.90)$$
$$\inf I_A(\lambda x_1 + (1-\lambda)x_2) \leq \max(\inf I_A(x_1), \inf I_A(x_2)), \quad (1.91)$$
$$\sup I_A(\lambda x_1 + (1-\lambda)x_2) \leq \max(\sup I_A(x_1), \sup I_A(x_2)), \quad (1.92)$$
$$\inf F_A(\lambda x_1 + (1-\lambda)x_2) \leq \max(\inf F_A(x_1), \inf F_A(x_2)), \quad (1.93)$$
$$\sup F_A(\lambda x_1 + (1-\lambda)x_2) \leq \max(\sup F_A(x_1), \sup F_A(x_2)), \quad (1.94)$$

for all x_1 and x_2 in X and all λ in $[0,1]$.

Theorem 6

If A and B are convex, so is their intersection.

Definition 28 (Strongly Convex)

An interval neutrosophic set A is strongly convex if for any two distinct points x_1 and x_2, and any λ in the open interval (0,1),

$$
\begin{aligned}
\inf T_A(\lambda x_1 + (1-\lambda)x_2) &> \min(\inf T_A(x_1), \inf T_A(x_2)), && (1.95)\\
\sup T_A(\lambda x_1 + (1-\lambda)x_2) &> \min(\sup T_A(x_1), \sup T_A(x_2)), && (1.96)\\
\inf I_A(\lambda x_1 + (1-\lambda)x_2) &< \max(\inf I_A(x_1), \inf I_A(x_2)), && (1.97)\\
\sup I_A(\lambda x_1 + (1-\lambda)x_2) &< \max(\sup I_A(x_1), \sup I_A(x_2)), && (1.98)\\
\inf F_A(\lambda x_1 + (1-\lambda)x_2) &< \max(\inf F_A(x_1), \inf F_A(x_2)), && (1.99)\\
\sup F_A(\lambda x_1 + (1-\lambda)x_2) &< \max(\sup F_A(x_1), \sup F_A(x_2)), && (1.100)
\end{aligned}
$$

for all x_1 and x_2 in X and all λ in [0,1].

Theorem 7

If A and B are strongly convex, so is their intersection.

3. Interval Neutrosophic Logic

Introduction

We present a novel interval neutrosophic logic that generalizes the interval valued fuzzy logic, the intuitionistic fuzzy logic and paraconsistent logics which only consider truth-degree or falsity-degree of a proposition.

In the interval neutrosophic logic, we consider not only truth-degree and falsity-degree but also indeterminacy-degree which can reliably capture more information under uncertainty.

We introduce mathematical definitions of an interval neutrosophic propositional calculus and an interval neutrosophic predicate calculus.

We propose a general method to design an interval neutrosophic logic system which consists of neutrosophication, neutrosophic inference, a neutrosophic rule base, neutrosophic type reduction and deneutrosophication.

A neutrosophic rule contains input neutrosophic linguistic variables and output neutrosophic linguistic variables.

A neutrosophic linguistic variable has neutrosophic linguistic values which defined by interval neutrosophic sets characterized by three membership functions: truth-membership, falsity-membership and indeterminacy-membership.

The interval neutrosophic logic can be applied to many potential real applications where information is imprecise, uncertain, incomplete and inconsistent such as Web intelligence, medical informatics, bioinformatics, decision making, etc.

Interval Neutrosophic Propositional Calculus

We introduce the elements of an interval neutrosophic propositional calculus based on the definition of the interval neutrosophic sets by using the notations from the theory of classical propositional calculus.

E. Mendelson, *Introduction to mathematical logic*, Van Nostrand, Princeton, NJ, 1987, Third edition.

Syntax of Interval Neutrosophic Propositional Calculus

Definition 29

An alphabet of the interval neutrosophic propositional calculus consists of three classes of symbols:

1. A set of interval neutrosophic propositional variables, denoted by lower-case letters, sometimes indexed;

2. Five connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$ which are called conjunction, disjunction, negation, implication, and biimplication symbols respectively;

3. The parentheses (and).

The alphabet of the interval neutrosophic propositional calculus has combinations obtained by assembling connectives and interval neutrosophic propositional variables in strings.

The purpose of the construction rules is to have the specification of distinguished combinations, called formulas.

Definition 30

The set of formulas (well-formed formulas) of interval neutrosophic propositional calculus is defined as follows.

1. Every interval neutrosophic propositional variable is a formula;
2. If p is a formula, then so is $(\neg p)$;
3. If p and q are formulas, then so are
   (a) $(p \wedge q)$,
   (b) $(p \vee q)$,
   (c) $(p \rightarrow q)$,
   (d) $(p \leftrightarrow q)$.
4. No sequence of symbols is a formula which is not required to be by 1, 2, and 3.

To avoid having formulas cluttered with parentheses, we adopt the following precedence hierarchy, with the highest precedence at the top:

$$\neg,$$

$$\wedge, \vee,$$

$$\rightarrow, \leftrightarrow.$$

Here is an example of the interval neutrosophic propositional calculus formula:

$$\neg p_1 \wedge p_2 \vee (p_1 \rightarrow p_3) \rightarrow p_2 \wedge \neg p_3$$

Definition 31

The language of interval neutrosophic propositional calculus given by an alphabet consists of the set of all formulas constructed from the symbols of the alphabet.

Semantics of Interval Neutrosophic Propositional Calculus

The study of interval neutrosophic propositional calculus comprises, among others, a syntax, which has the distinction of well-formed formulas, and a semantics, the purpose of which is the assignment of a meaning to well-formed formulas.

To each interval neutrosophic proposition p, we associate it with an ordered triple components

$$<t(p), i(p), f(p)>,$$

where $t(p), i(p), f(p) \subseteq [0,1]$.

$t(p), i(p), f(p)$ are called truth-degree, indeterminacy-degree and falsity-degree of p.

Let this assignment be provided by an interpretation function or interpretation INL defined over a set of propositions P in such a way that

$$\mathrm{INL}(p) = <t(p), i(p), f(p)> .$$

Hence, the function INL : $P \to N$ gives the truth, indeterminacy and falsity degrees of all propositions in P.

We assume that the interpretation function INL assigns to the logical truth T, and to F :

$$\text{INL}(T) = <1, 0, 0>,$$
$$\text{INL}(F) = <0, 1, 1>.$$

An interpretation which makes a formula true is a model of the formula.

Let i, l be the subinterval of $[0,1]$.

Then

$$i + l = [\inf i + \inf l, \sup i + \sup l],$$

$$i - l = [\inf i - \sup l, \sup i - \inf l],$$

$$\max(i, l) = [\max(\inf i, \inf l), \max(\sup i, \sup l)],$$

$$\min(i, l) = [\min(\inf i, \inf l), \min((\sup i, \sup l))].$$

The semantics of four interval neutrosophic propositional connectives is given in *Table* below.

Note that p ↔ q if and only if p → q and q → p.

Connectives	Semantics
$INL(\neg p)$	$\langle f(p), 1 - i(p), t(p) \rangle$
$INL(p \wedge q)$	$\langle \min(t(p), t(q)), \max(i(p), i(q)), \max(f(p), f(q)) \rangle$
$INL(p \vee q)$	$\langle \max(t(p), t(q)), \min(i(p), i(q)), \min(f(p), f(q)) \rangle$
$INL(p \rightarrow q)$	$\langle \min(1, 1 - t(p) + t(q)), \max(0, i(q) - i(p)), \max(0, f(q) - f(p)) \rangle$

Table 1: Semantics of Four Connectives in Interval Neutrosophic Propositional Logic.

Example 10

$$\text{INL}(p) = <0.5, 0.4, 0.7>,$$
$$\text{INL}(q) = <1, 0.7, 0.2>.$$

Then,

$$\text{INL}(\neg p) = < 0.7, 0.6, 0.5>,$$

$$\text{INL}(p \wedge \neg p) = <0.5, 0.4, 0.7>,$$

$$\text{INL}(p \vee q) = <1, 0.7, 0.2>,$$

$$\text{INL}(p \rightarrow q) = <1, 1, 0>.$$

A given well-formed interval neutrosophic propositional formula will be called a tautology (valid) if $\text{INL}(A) = <1, 1, 0>$, for all interpretation functions INL.

It will be called a contradiction (inconsistent) if $\text{INL}(A) = <0, 0, 1>$, for all interpretation functions INL.

Definition 32

Two formulas p and q are said to be equivalent, denoted $p = q$, if and only if the

$$\mathrm{INL}(p) = \mathrm{INL}(q)$$

for every interpretation function INL.

Theorem 8

Let F be the set of formulas and ∧ be the meet and ∨ the join, then $<F; \wedge, \vee>$ is a distributive lattice.

Theorem 9

If p and $p \rightarrow q$ are tautologies, then q is also a tautology.

Proof Since p and $p \rightarrow q$ are tautologies then for every INL, $INL(p) = INL(p \rightarrow q) = \langle 1, 0, 0\rangle$, that is $t(p) = 1, i(p) = f(p) = 0$, $t(p \rightarrow q) = \min(1, 1 - t(p) + t(q)) = 1$, $i(p \rightarrow q) = \max(0, i(q) - f(p)) = 0$, $f(p \rightarrow q) = \max(0, f(q) - f(p)) = 0$. Hence,
$\mathrm{t(q)} = 1$, $\mathrm{i(q)} = \mathrm{f(q)} = 0$. So q is a tautology.

Proof Theory of Interval Neutrosophic Propositional Calculus

We give the proof theory for interval neutrosophic propositional logic to complement the semantics part.

Definition 33

The interval neutrosophic propositional logic is defined by the following axiom schema.

$$p \to (q \to p)$$
$$p_1 \wedge \ldots \wedge p_m \to q_1 \vee \ldots q_n \textit{ provided some } p_i \textit{ is some } q_j$$
$$p \to (q \to p \wedge q)$$
$$(p \to r) \to ((q \to r) \to (p \vee q \to r))$$
$$(p \vee q) \to r \textit{ iff } p \to r \textit{ and } q \to r$$
$$p \to q \textit{ iff } \neg q \to \neg p$$
$$p \to q \textit{ and } q \to r \textit{ implies } p \to r$$
$$p \to q \textit{ iff } p \leftrightarrow (p \wedge q) \textit{ iff } q \to (p \vee q)$$

The concept of (formal) deduction of a formula from a set of formulas, that is, using the standard notation, $\Gamma \vdash p$, is defined as usual; in this case, we say that p is a syntactical consequence of the formulas in T.

Theorem 10

For interval neutrosophic propositional logic, we have:

1. $\{p\} \vdash p$,
2. $\Gamma \vdash p$ *entails* $\Gamma \cup \Delta \vdash p$,
3. *if* $\Gamma \vdash p$ *for any* $p \in \Delta$ *and* $\Delta \vdash q$, *then* $\Gamma \vdash q$.

Theorem 11

In interval neutrosophic propositional logic, we have:

1. $\neg\neg p \leftrightarrow p$
2. $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$
3. $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$

Theorem 12

In interval neutrosophic propositional logic, the following schema do not hold:

1. $p \vee \neg p$
2. $\neg(p \wedge \neg p)$
3. $p \wedge \neg p \rightarrow q$
4. $p \wedge \neg p \rightarrow \neg q$
5. $\{p, p \rightarrow q\} \vdash q$
6. $\{p \rightarrow q, \neg q\} \vdash \neg p$
7. $\{p \vee q, \neg q\} \vdash p$
8. $\neg p \vee q \leftrightarrow p \rightarrow q$

Example 11

To illustrate the use of the interval neutrosophic propositional consequence relation, let's consider the following example.

$$p \rightarrow (q \wedge r)$$
$$r \rightarrow s$$
$$q \rightarrow \neg s$$
$$a$$

From $p \rightarrow (q \wedge r)$, we get $p \rightarrow q$ and $p \rightarrow r$.

From $p \rightarrow q$ and $q \rightarrow \neg s$, we get $p \rightarrow \neg s$.

From $p \rightarrow r$ and $r \rightarrow s$, we get $p \rightarrow s$.

Hence, p is equivalent to $p \wedge s$ and $p \wedge \neg s$.

However, we cannot detach s from p nor $\neg s$ from p.

This is in part due to interval neutrosophic propositional logic incorporating neither modus ponens nor and elimination.

Interval Neutrosophic Predicate Calculus

We extend our consideration to the full language of first order interval neutrosophic predicate logic. First we give the formalization of syntax of first order interval neutrosophic predicate logic as in classical first-order predicate logic.

Syntax of Interval Neutrosophic Predicate Calculus

Definition 34

An alphabet of the first order interval neutrosophic predicate calculus consists of seven classes of symbols:

1. variables, denoted by lower-case letters, sometimes indexed;
2. constants, denoted by lower-case letters;
3. function symbols, denoted by lower-case letters, sometimes indexed;
4. predicate symbols, denoted by lower-case letters, sometimes indexed;
5. five connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$ which are called the conjunction, disjunction, negation, implication, and biimplication symbols respectively;
6. two quantifiers, the universal quantifier $\forall$ (for all) and the existential quantifier $\exists$ (there exists);
7. the parentheses (and).

To avoid having formulas cluttered with brackets, we adopt the following precedence hierarchy, with the highest precedence at the top:

$$\neg, \forall, \exists$$
$$\wedge, \vee$$
$$\rightarrow, \leftrightarrow$$

Next we turn to the definition of the first order interval neutrosophic language given by an alphabet.

Definition 35

A term is defined as follows:

1. A variable is a term.
2. A constant is a term.
3. If f is a n-ary function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1,\ldots,f_n)$ is a term.

Definition 36

A (well-formed) formula is defined inductively as follows:

1. If p is a n-ary predicate symbol and $t_1, ..., t_n$ are terms, then $p(t_1, ..., t_n)$ is a formula (called an atomic formula or, more simply, an atom).
2. If F and G are formulas, then so are $(\neg F)$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$ and $(F \leftrightarrow G)$.
3. If F is a formula and x is a variable, then $(\forall xF)$ and $(\exists xF)$ are formulas.

Definition 37

The first order interval neutrosophic language given by an alphabet consists of the set of all formulas constructed from the symbols of the alphabet.

Example 12

$\forall x \exists y(p(x,y) \rightarrow q(x)), \neg \exists x(p(x,a) \wedge q(x))$ are formulas.

Definition 38

The scope of $\forall x$ (resp. $\exists x$) in $\forall xF$ (resp. $\exists xF$) is F.

A bound occurrence of a variable in a formula is an occurrence immediately following a quantifier or an occurrence within the scope of a quantifier, which has the same variable immediately after the quantifier.

Any other occurrence of a variable is free.

Example 13

In the formula $\forall x p(x,y) \vee q(x)$, the first two occurrences of x are bound, while the third occurrence is free, since the scope of $\forall x$ is $p(x, y)$ and y is free

Semantics of Interval Neutrosophic Predicate Calculus

We study the semantics of interval neutrosophic predicate calculus, the purpose of which is the assignment of a meaning to well-formed formulas.

In the interval neutrosophic propositional logic, an interpretation is an assignment of truth values (ordered triple component) to propositions.

In the first order interval neutrosophic predicate logic, since there are variables involved, we have to do more than that.

To define an interpretation for a well-formed formula in this logic, we have to specify two things, the domain and an assignment to constants and predicate symbols occurring in the formula.

The following is the formal definition of an interpretation of a formula in the first order interval neutrosophic predicate logic.

Definition 39

An interpretation function (or interpretation) of a formula F in the first order interval neutrosophic predicate logic consists of a nonempty domain D, and an assignment of "values" to each constant and predicate symbol occurring in F as follows:

1. To each constant, we assign an element in D.

2. To each n-ary function symbol, we assign a mapping from D^n to D. (Note that $D^n = \{(x_1, ..., x_n) \mid x_1 \in D, ..., x_n \in D\}$).

3. Predicate symbols get their meaning through evaluation functions E which assign to each variable x an element $E(x) \in D$.

To each n-ary predicate symbol p, there is a function INP(p) : $D^n \to N$. So:

$$INP(p(x_1, \ldots, x_n)) = INP(p)(E(x_1), \ldots, E(x_n)).$$

That is,

$$INP(p)(a_1, \ldots, a_n) = \langle t(p(a_1, \ldots, a_n)), i(p(a_1, \ldots, a_n)), f(p(a_1, \ldots, a_n)),$$

where

$$t(p(a_1, \ldots, a_n)), i(p(a_1, \ldots, a_n)), f(p(a_1, \ldots, a_n)) \subseteq [0, 1].$$

They are called truth-degree, indeterminacy degree and falsity-degree of $p(a_1, ..., a_n)$ respectively.

We assume that the interpretation function INP assigns to the logical truth

$$T : \text{INP}(T) = <1, 1, 0>,$$

and to

$$F : \text{INP}(F) = <0, 0, 1>.$$

The semantics of four interval neutrosophic predicate connectives and two quantifiers is given in *Table* next slide.

For simplification of notation, we use p to denote $p(a_1, ..., a_i)$.

Note that $p \leftrightarrow q$ if and only if $p \rightarrow q$ and $q \rightarrow p$.

Connectives	Semantics
$INP(\neg p)$	$\langle f(p), 1-i(p), t(p)\rangle$
$INP(p \wedge q)$	$\langle \min(t(p), t(q)), \max(i(p), i(q)), \max(f(p), f(q))\rangle$
$INP(p \vee q)$	$\langle \max(t(p), t(q)), \min(i(p), i(q)), \min(f(p), f(q))\rangle$
$INP(p \rightarrow q)$	$\langle \min(1, 1-t(p)+t(q)), \max(0, i(q)-i(p)), \max(0, f(q)-f(p))\rangle$
$INP(\forall x F)$	$\langle \min t(F(E(x))), \min i(F(E(x))), \max f(F(E(x)))\rangle,\ E(x) \in D$
$INP(\exists x F)$	$\langle \max t(F(E(x))), \max i(F(E(x))), \min f(F(E(x)))\rangle,\ E(x) \in D$

Table 2: Semantics of Four Connectives and Two Quantifiers in Interval Neutrosophic Predicate Logic.

Definition 40

A formula *F* is consistent (satisfiable) if and only if there exists an interpretation *I* such that *F* is evaluated to <1, 1, 0> in *I*. If a formula *F* is *T* in an interpretation *I*, we say that *I* is a model of *F* and *I* satisfies *F*.

Definition 41

A formula *F* is inconsistent (unsatisfiable) if and only if there exists no interpretation that satisfies *F*.

Definition 42

A formula F is valid if and only if every interpretation of F satisfies F.

Definition 43

A formula F is a logical consequence of formulas F_1, ..., F_n if and only if for every interpretation I, if $F_1 \wedge \ldots \wedge F_n$ is true in I, F is also true in I.

Example 15

$(\forall x)(p(x) \rightarrow (\exists y)p(y))$ *is valid,* $(\forall x)p(x) \wedge (\exists y)\neg p(y)$ *is consistent.*

Theorem 13

There is no inconsistent formula in the first order interval neutrosophic predicate logic.

Proof Theory of Interval Neutrosophic Predicate Calculus

We give the proof theory for first order interval neutrosophic predicate logic to complement the semantics part.

Definition 44

The first order interval neutrosophic predicate logic is defined by the following axiom schema.

$$(p \rightarrow q(x)) \rightarrow (p \rightarrow \forall x q(x))$$
$$\forall x p(x) \rightarrow p(a)$$
$$p(x) \rightarrow \exists x p(x)$$
$$(p(x) \rightarrow q) \rightarrow (\exists x p(x) \rightarrow q)$$

Theorem 14

In the first order interval neutrosophic predicate logic, we have:

1. $p(x) \vdash \forall x p(x)$
2. $p(a) \vdash \exists x p(x)$
3. $\forall x p(x) \vdash p(y)$
4. $\Gamma \cup \{p(x)\} \vdash q,\ then\ \Gamma \cup \{\exists x p(x)\} \vdash q$

Theorem 15

In the first order interval neutrosophic predicate logic, the following schemes are valid, where r is a formula in which x does not appear free:

1. $\forall x r \leftrightarrow r$
2. $\exists x r \leftrightarrow r$
3. $\forall x \forall y p(x,y) \leftrightarrow \forall y \forall x p(x,y)$
4. $\exists x \exists y p(x,y) \leftrightarrow \exists y \exists x p(x,y)$
5. $\forall x \forall y p(x,y) \rightarrow \forall x p(x,x)$
6. $\exists x p(x,x) \rightarrow \exists x \exists y p(x,y)$
7. $\forall x p(x) \rightarrow \exists x p(x)$
8. $\exists x \forall y p(x,y) \rightarrow \forall y \exists x p(x,y)$
9. $\forall x (p(x) \wedge q(x)) \leftrightarrow \forall x p(x) \wedge \forall x q(x)$
10. $\exists x (p(x) \vee q(x)) \leftrightarrow \exists x p(x) \vee \exists x q(x)$
11. $p \wedge \forall x q(x) \leftrightarrow \forall x (p \wedge q(x))$
12. $p \vee \forall x q(x) \leftrightarrow \forall x (p \vee q(x))$
13. $p \wedge \exists x q(x) \leftrightarrow \exists x (p \wedge q(x))$
14. $p \vee \exists x q(x) \leftrightarrow \exists x (p \vee q(x))$

15. $\forall x(p(x) \to q(x)) \to (\forall x p(x) \to \forall x q(x))$

16. $\forall x(p(x) \to q(x)) \to (\exists x p(x) \to \exists x q(x))$

17. $\exists x(p(x) \land q(x)) \to \exists x p(x) \land \exists x q(x)$

18. $\forall x p(x) \lor \forall x q(x) \to \forall x(p(x) \lor q(x))$

19. $\neg\exists x \neg p(x) \leftrightarrow \forall x p(x)$

20. $\neg\forall x \neg p(x) \leftrightarrow \exists p(x)$

21. $\neg\exists x p(x) \leftrightarrow \forall x \neg p(x)$

22. $\exists x \neg p(x) \leftrightarrow \neg\forall x p(x)$

An Application of Interval Neutrosophic Logic

We provide one practical application of the interval neutrosophic logic – Interval Neutrosophic Logic System (INLS).

INLS can handle rule uncertainty as same as type-2 FLS, besides, it can handle rule inconsistency without the danger of trivialization.

Like the classical FLS, INLS is also characterized by IF-THEN rules.

INLS consists of neutrosophication, neutrosophic inference, a neutrosophic rule base, neutrosophic type reduction and deneutrosophication.

Given an input vector $x = (x_1, ..., x_n)$, where $x_1, ..., x_n$ can be crisp inputs or neutrosophic sets, the INLS will generate a crisp output y.

The general scheme of INLS is shown in *Figure* next slide.

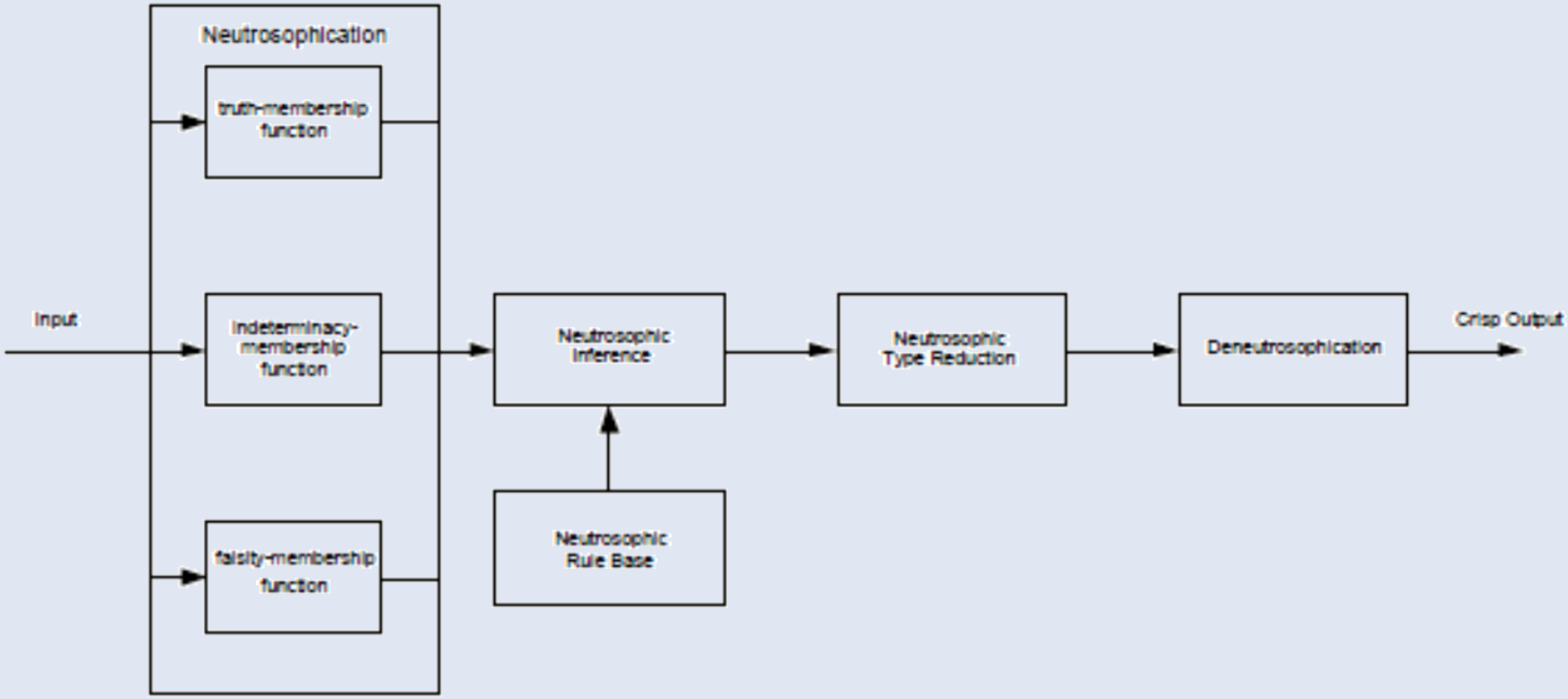


Figure 3: General Scheme of an INLS.

For implication $p \rightarrow q$, we define the semantics as:

$$\sup t_{p\rightarrow q} = \min(\sup t(p), \sup t(q)), \quad (2.1)$$
$$\inf t_{p\rightarrow q} = \min(\inf t(p), \inf t(q)), \quad (2.2)$$
$$\sup i_{p\rightarrow q} = \max(\sup i(p), \sup i(q)), \quad (2.3)$$
$$\inf i_{p\rightarrow q} = \max(\inf i(p), \inf i(q)), \quad (2.4)$$
$$\sup f_{p\rightarrow q} = \max(\sup f(p), \sup f(q)), \quad (2.5)$$
$$\inf f_{p\rightarrow q} = \max(\inf f(p), \inf f(q)), \quad (2.6)$$

where $t(p)$, $i(p)$, $f(p)$, $t(q)$, $i(q)$, $f(q) \subseteq [0,1]$.

Let $X = X_1 \times \cdots \times X_n$.

The truth-membership function, indeterminacy-membership function and falsity-membership function of a fired k^{th} rule can be represented using the definition of interval neutrosophic composition functions and the semantics of conjunction and disjunction defined in *Table* next slide and equations (2.1–2.6) as:

$$\sup T_{\tilde{B}^k}(y) = \sup_{x\in X} \min(\sup T_{\tilde{A}_1^k}(x_1), \sup T_{A_1^k}(x_1), \ldots, \sup T_{\tilde{A}_n^k}(x_n), \sup T_{A_n^k}(x_n), \sup T_{B^k}(y)), \quad (2.7)$$

$$\inf T_{\tilde{B}^k}(y) = \sup_{x\in X} \min(\inf T_{\tilde{A}_1^k}(x_1), \inf T_{A_1^k}(x_1), \ldots, \inf T_{\tilde{A}_n^k}(x_n), \inf T_{A_n^k}(x_n), \inf T_{B^k}(y)), \quad (2.8)$$

$$\sup I_{\tilde{B}^k}(y) = \sup_{x\in X} \max(\sup I_{\tilde{A}_1^k}(x_1), \sup I_{A_1^k}(x_1), \ldots, \sup I_{\tilde{A}_n^k}(x_n), \sup I_{A_n^k}(x_n), \sup I_{B^k}(y)), \quad (2.9)$$

$$\inf I_{\tilde{B}^k}(y) = \sup_{x\in X} \max(\inf I_{\tilde{A}_1^k}(x_1), \inf I_{A_1^k}(x_1), \ldots, \inf I_{\tilde{A}_n^k}(x_n), \inf I_{A_n^k}(x_n), \inf I_{B^k}(y)), \quad (2.10)$$

$$\sup F_{\tilde{B}^k}(y) = \inf_{x\in X} \max(\sup F_{\tilde{A}_1^k}(x_1), \sup F_{A_1^k}(x_1), \ldots, \sup F_{\tilde{A}_n^k}(x_n), \sup F_{A_n^k}(x_n), \sup F_{B^k}(y)), \quad (2.11)$$

$$\inf F_{\tilde{B}^k}(y) = \inf_{x\in X} \max(\inf F_{\tilde{A}_1^k}(x_1), \inf F_{A_1^k}(x_1), \ldots, \inf F_{\tilde{A}_n^k}(x_n), \inf F_{A_n^k}(x_n), \inf F_{B^k}(y)), \quad (2.12)$$

Figure below shows the conceptual diagram for neutrosophication of a crisp input x_1.

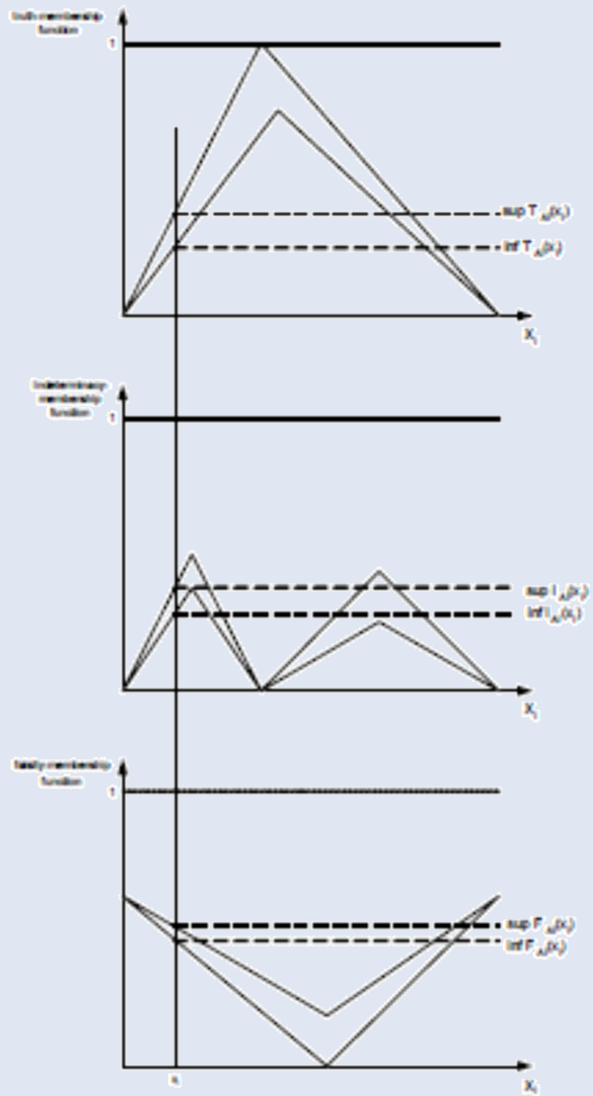


Figure 4: Conceptual Diagram for Neutrosophication of Crisp Input

We give the algorithmic description of INLS.

BEGIN

Step 1: Neutrosophication

The purpose of neutrosophication is to map inputs into interval neutrosophic input sets. Let G_i^k be an interval neutrosophic input set to represent the result of neutrosophication of i^{th} input variable of k^{th} rule, then

$$\sup T_{G_i^k}(x_i) = \sup_{x_i \in X_i} \min(\sup T_{\tilde{A}_i^k}(x_i), \sup T_{A_i^k}(x_i)), \tag{2.13}$$

$$\inf T_{G_i^k}(x_i) = \sup_{x_i \in X_i} \min(\inf T_{\tilde{A}_i^k}(x_i), \inf T_{A_i^k}(x_i)), \tag{2.14}$$

$$\sup I_{G_i^k}(x_i) = \sup_{x_i \in X_i} \max(\sup I_{\tilde{A}_i^k}(x_i), \sup I_{A_i^k}(x_i)), \tag{2.15}$$

$$\inf I_{G_i^k}(x_i) = \sup_{x_i \in X_i} \max(\inf I_{\tilde{A}_i^k}(x_i), \inf I_{A_i^k}(x_i)), \tag{2.16}$$

$$\sup F_{G_i^k}(x_i) = \inf_{x_i \in X_i} \max(\sup F_{\tilde{A}_i^k}(x_i), \sup F_{A_i^k}(x_i)), \tag{2.17}$$

$$\inf F_{G_i^k}(x_i) = \inf_{x_i \in X_i} \max(\inf F_{\tilde{A}_i^k}(x_i), \inf F_{A_i^k}(x_i)), \tag{2.18}$$

where $x_i \in X_i$.

If x_i are crisp inputs, then equations (50–55) are simplified to

$$\sup T_{G_i^k}(x_i) = \sup T_{A_i^k}(x_i), \quad (2.19)$$
$$\inf T_{G_i^k}(x_i) = \inf T_{A_i^k}(x_i), \quad (2.20)$$
$$\sup I_{G_i^k}(x_i) = \sup I_{A_i^k}(x_i), \quad (2.21)$$
$$\inf I_{G_i^k}(x_i) = \inf I_{A_i^k}(x_i), \quad (2.22)$$
$$\sup F_{G_i^k}(x_i) = \sup F_{A_i^k}(x_i), \quad (2.23)$$
$$\inf F_{G_i^k}(x_i) = \inf F_{A_i^k}(x_i), \quad (2.24)$$

where $x_i \in X_i$.

Step 2: Neutrosophic Inference

The core of INLS is the neutrosophic inference, the principle of which has already been explained above. Suppose the k^{th} rule is fired. Let G^k be an interval neutrosophic set to represent the result of the input and antecedent operation for k^{th} rule, then

$$\sup T_{G^k}(x) = \sup_{x\in X} \min(\sup T_{\bar{A}_1^k}(x_1), \sup T_{A_1^k}(x_1), \ldots, \sup T_{\bar{A}_n^k}(x_n), \sup T_{A_n^k}(x_n)), \quad (2.25)$$
$$\inf T_{G^k}(x) = \sup_{x\in X} \min(\inf T_{\bar{A}_1^k}(x_1), \inf T_{A_1^k}(x_1), \ldots, \inf T_{\bar{A}_n^k}(x_n), \inf T_{A_n^k}(x_n)), \quad (2.26)$$
$$\sup I_{G^k}(x) = \sup_{x\in X} \max(\sup I_{\bar{A}_1^k}(x_1), \sup I_{A_1^k}(x_1), \ldots, \sup I_{\bar{A}_n^k}(x_n), \sup I_{A_n^k}(x_n)), \quad (2.27)$$
$$\inf I_{G^k}(x) = \sup_{x\in X} \max(\inf I_{\bar{A}_1^k}(x_1), \inf I_{A_1^k}(x_1), \ldots, \inf I_{\bar{A}_n^k}(x_n), \inf I_{A_n^k}(x_n)), \quad (2.28)$$
$$\sup F_{G^k}(x) = \inf_{x\in X} \max(\sup F_{\bar{A}_1^k}(x_1), \sup F_{A_1^k}(x_1), \ldots, \sup F_{\bar{A}_n^k}(x_n), \sup F_{A_n^k}(x_n)), \quad (2.29)$$
$$\inf F_{G^k}(x) = \inf_{x\in X} \max(\inf F_{\bar{A}_1^k}(x_1), \inf F_{A_1^k}(x_1), \ldots, \inf F_{\bar{A}_n^k}(x_n), \inf F_{A_n^k}(x_n)), \quad (2.30)$$

where $x_i \in X_i$.

Here we restate the result of neutrosophic inference:

$$\sup T_{\bar{B}^k}(y) = \min(\sup T_{G_1^k}(x), \sup T_{B^k}(y)), \tag{2.31}$$

$$\inf T_{\bar{B}^k}(y) = \min(\inf T_{G^k}(x), \inf T_{B^k}(y)), \tag{2.32}$$

$$\sup I_{\bar{B}^k}(y) = \max(\sup I_{G^k}(x), \sup I_{B^k}(y)), \tag{2.33}$$

$$\inf I_{\bar{B}^k}(y) = \max(\inf I_{G^k}(x), \inf I_{B^k}(y)), \tag{2.34}$$

$$\sup F_{\bar{B}^k}(y) = \max(\sup F_{G^k}(x), \sup F_{B^k}(y)), \tag{2.35}$$

$$\inf F_{\bar{B}^k}(y) = \max(\inf F_{G^k}(x), \inf F_{B^k}(y)), \tag{2.36}$$

where $x \in X, y \in Y$.

Suppose that N rules in the neutrosophic rule base are fired, where $N \leq M$, then, the output interval neutrosophic set $\check{B}$ is:

$$\sup T_{\tilde{B}}(y) = \max_{k=1}^{N} \sup T_{\tilde{B}^k}(y), \tag{2.37}$$

$$\inf T_{\tilde{B}}(y) = \max_{k=1}^{N} \inf T_{\tilde{B}^k}(y), \tag{2.38}$$

$$\sup I_{\tilde{B}}(y) = \min_{k=1}^{N} \sup I_{\tilde{B}^k}(y), \tag{2.39}$$

$$\inf I_{\tilde{B}}(y) = \min_{k=1}^{N} \inf I_{\tilde{B}^k}(y), \tag{2.40}$$

$$\sup F_{\tilde{B}}(y) = \min_{k=1}^{N} \sup T_{\tilde{B}^k}(y), \tag{2.41}$$

$$\inf T_{\tilde{B}}(y) = \min_{k=1}^{N} \inf T_{\tilde{B}^k}(y), \tag{2.42}$$

where $y \in \Upsilon$.

Step 3: Neutrosophic type reduction

We do the neutrosophic type reduction to transform each interval into one number. There are many ways to do it, here, we give one method:

$$T'_{\tilde{B}}(y) = (\inf T_{\tilde{B}}(y) + \sup T_{\tilde{B}}(y))/2, \quad (2.43)$$

$$I'_{\tilde{B}}(y) = (\inf I_{\tilde{B}}(y) + \sup I_{\tilde{B}}(y))/2, \quad (2.44)$$

$$F'_{\tilde{B}}(y) = (\inf F_{\tilde{B}}(y) + \sup F_{\tilde{B}}(y))/2, \quad (2.45)$$

where $y \in \Upsilon$.

So, after neutrosophic type reduction, we will get an ordinary neutrosophic set (a type-1 neutrosophic set) $\check{B}$.

Then we need to do the deneutrosophication to get a crisp output.

Step 4: Deneutrosophication

The purpose of deneutrosophication is to convert an ordinary neutrosophic set (a type-1 neutrosophic set) obtained by neutrosophic type reduction to a single real number which represents the real output.

Similar to defuzzification, there are many deneutrosophication methods according to different applications.

Here we give one method.

The deneutrosophication process consists of two steps.

Step 4.1: Synthesization

It is the process to transform an ordinary neutrosophic set (a type-1 neutrosophic set) $\check{B}$ into a fuzzy set $\bar{B}$.

It can be expressed using the following function:

$$f(T'_{\check{B}}(y), I'_{\check{B}}(y), F'_{\check{B}}(y)) : [0,1] \times [0,1] \times [0,1] \to [0,1] \quad (2.46)$$

Here we give one definition of *f*:

$$T_{\bar{B}}(y) = a * T'_{\bar{B}}(y) + b * (1 - F'_{\bar{B}}(y)) + c * I'_{\bar{B}}(y)/2 + d * (1 - I'_{\bar{B}}(y)/2), \quad (2.47)$$

where $0 \leq a, b, c, d \leq 1$, $a + b + c + d = 1$.

The purpose of synthesization is to calculate the overall truth degree according to three components: truth-membership function, indeterminacy-membership function and falsity-membership function.

The component–truth-membership function gives the direct information about the truth-degree, so we use it directly in the formula.

The component–falsity-membership function gives the indirect information about the truth-degree, so we use (1−F) in the formula.

To understand the meaning of indeterminacy-membership function I, we give an example: a statement is "The quality of service is good", now firstly a person has to select a decision among {T, I, F}, secondly he or she has to answer the degree of the decision in [0,1].

If he or she chooses $I = 1$, it means 100% "not sure" about the statement, i.e., 50% true and 50% false for the statement (100% balanced), in this sense, $I = 1$ contains the potential truth value 0.5.

If he or she chooses $I = 0$, it means 100% "sure" about the statement, i.e., either 100% true or 100% false for the statement (0% balanced), in this sense, $I = 0$ is related to two extreme cases, but we do not know which one is in his or her mind.

So we have to consider both at the same time: $I = 0$ contains the potential truth value that is either 0 or 1.

If I decreases from 1 to 0, then the potential truth value changes from one value 0.5 to two different possible values gradually to the final possible ones 0 and 1 (i.e., from 100% balanced to 0% balanced), since he or she does not choose either T or F but I, we do not know his or her final truth value.

Therefore, the formula has to consider two potential truth values implicitly represented by I with different weights (c and d) because of lack of his or her final decision information after he or she has chosen I.

Generally, a > b > c,d; c and d could be decided subjectively or objectively as long as enough information is available. The parameters a,b,c and d can be tuned using learning algorithms such as neural networks and genetic algorithms in the development of application to improve the performance of the INLS.

Generally, $a > b > c, d$; c and d could be decided subjectively or objectively as long as enough information is available.

The parameters a, b, c and d can be tuned using learning algorithms such as neural networks and genetic algorithms in the development of application to improve the performance of the INLS.

Step 4.2: Calculation of a typical neutrosophic value

Here we introduce one method of calculation of center of area.

The method is sometimes called the center of gravity method or centroid method, the deneutrosophicated value, $dn(T_{\bar{B}}(y))$ is calculated by the formula

$$dn(T_{\bar{B}}(y)) = \frac{\int_{\alpha}^{\beta} T_B(y)y\,dy}{\int_{\alpha}^{\beta} T_B\,dy}. \qquad (2.48)$$

END

Conclusions

We gave the formal definitions of interval neutrosophic logic which are extensions of many other classical logics such as fuzzy logic, intuitionistic fuzzy logic and paraconsistent logics.

Interval neutrosophic logic includes interval neutrosophic propositional logic and first order interval neutrosophic predicate logic. We call them classical (standard) neutrosophic logic.

In the future, we also will discuss and explore the non-classical (non-standard) neutrosophic logic such as modal interval neutrosophic logic, temporal interval neutrosophic logic, etc.

Interval neutrosophic logic can not only handle imprecise, fuzzy and incomplete propositions but also inconsistent propositions without the danger of trivialization.

We gave one application based on the semantic notion of interval neutrosophic logic – the Interval Neutrosophic Logic Systems (INLS), which is the generalization of classical FLS and interval valued fuzzy FLS.

Interval neutrosophic logic will have a lot of potential applications in computational Web intelligence.

Y.-Q. Zhang, A. Kandel, T.Y. Lin, and Y.Y. Yao, *Computational web intelligence: Intelligent technology for web applications, series in machine perception and artificial intelligence*, World Scientific, 2004, Volume 58.

For example, current fuzzy Web intelligence techniques can be improved by using more reliable interval neutrosophic logic methods because T, I and F are all used in decision making.

In large, such robust interval neutrosophic logic methods can also be used in other applications such as medical informatics, bioinformatics and human-oriented decision-making under uncertainty.

In fact, interval neutrosophic sets and interval neutrosophic logic could be applied in the fields that fuzzy sets and fuzzy logic are suitable for, also the fields that paraconsistent logics are suitable for.

4. Neutrosophic Relational Data Model

Introduction

We present a generalization of the relational data model based on interval neutrosophic sets.

Our data model is capable of manipulating incomplete as well as inconsistent information.

Fuzzy relation or intuitionistic fuzzy relation can only handle incomplete information.

Associated with each relation are two membership functions one is called truth-membership function T which keeps track of the extent to which we believe the tuple is in the relation, another is called falsity-membership function which keeps track of the extent to which we believe that it is not in the relation.

A neutrosophic relation is inconsistent if there exists one tuple a such that

$$T(a) + F(a) > 1.$$

In order to handle inconsistent situation, we propose an operator called "split" to transform inconsistent neutrosophic relations into pseudo-consistent neutrosophic relations and do the set-theoretic and relation-theoretic operations on them and finally use another operator called "combine" to transform the result back to neutrosophic relation.

For this model, we define algebraic operators that are generalisations of the usual operators such as interesection, union, selection, join on fuzzy relations.

Our data model can underlie any database and knowledge-base management system that deals with incomplete and inconsistent information.

Relational data model was proposed by Ted Codd's pioneering paper.

E.F. Codd, *A relational model for large shared data banks*, Communications of the ACM **13** (1970), no. 6, 377–387.

Since then, relational database systems have been extensively studied and a lot of commercial relational database systems are currently available.

Elmasri and Navathe, *Fundamentals of database systems*, third ed., Addison–Wesley, New York, 2000.

A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database system concepts*, third ed., McGraw–Hill, Boston, 1996.

We present a new relational data model – neutrosophic relational data model (NRDM).

Our model is based on the neutrosophic set theory which is an extension of intuitionistic fuzzy set theory and is capable of manipulating incomplete as well as inconsistent information.

We use both truth-membership function grade α and falsity-membership function grade β to denote the status of a tuple of a certain relation with $\alpha, \beta \in [0,1]$ and $\alpha + \beta \leq 2$.

NRDM is the generalization of fuzzy relational data model (FRDM).

That is, when $\alpha + \beta = 1$, neutrosophic relation is the ordinary fuzzy relation.

This model is distinct with paraconsistent relational data model (PRDM), in fact it can be easily shown that PRDM is a special case of PIFRDM.

That is when $\alpha,\beta = 0$ or 1, neutrosophic relation is just paraconsistent relation.

We use *Figure* next slide to express the relationship among FRDM, PRDM and PIFRDM.

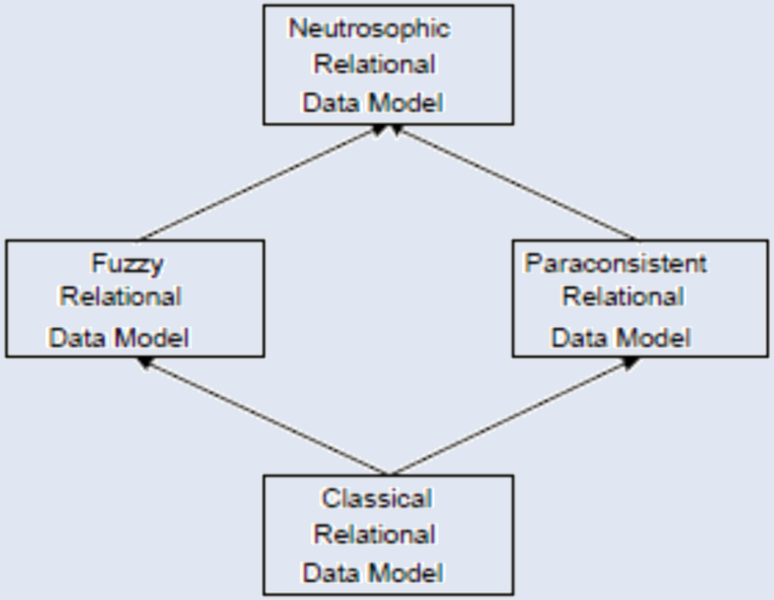


Figure 5: Relationship Among FRDM, PRDM, NRDM and RDM.

We introduce neutrosophic relations, which are the fundamental mathematical structures underlying our model.

These structures are strictly more general than classical fuzzy relations and intuitionistic fuzzy relations (interval-valued fuzzy relations), in that for any fuzzy relation or intuitionistic fuzzy relation (interval-valued fuzzy relation) there is a neutrosophic relation with the same information content, but not vice versa.

The claim is also true for the relationship between neutrosophic relations and paraconsistent relations.

We define algebraic operators over neutrosophic relations that extend the standard operators such as selection, join, union over fuzzy relations.

There are many potential applications of our new data model. Here are some examples:

a. Web mining. Essentially the data and documents on the Web are heterogeneous, inconsistency is unavoidable. Using the presentation and reasoning method of our data model, it is easier to capture imperfect information on the Web which will provide more potentially value-added information.
b. Bioinformatics. There is a proliferation of data sources. Each research group and each new experimental technique seems to generate yet another source of valuable data. But these data can be incomplete and imprecise and even inconsistent. We could not simply throw away one data in favor of other data. So how to represent and extract useful information from these data will be a challenge problem.
c. Decision Support System. In decision support system, we need to combine the database with the knowledge base. There will be a lot of uncertain and inconsistent information, so we need an efficient data model to capture these information and reasoning with these information.

Fuzzy Relations and Operations

We present the essential concepts of a fuzzy relational database.

Fuzzy relations associate a value between 0 and 1 with every tuple representing the degree of membership of the tuple in the relation.

We also present several useful query operators on fuzzy relations.

Let a relation scheme (or just scheme) Σ be a finite set of attribute names, where for any attribute name $A \in \Sigma$, $dom(A)$ is a non-empty domain of values for A.

A tuple on Σ is any map $t : \Sigma \rightarrow \cup_{A \in} dom(A)$, such that $t(A) \in dom(A)$, for each $A \in \Sigma$.

Let $\tau(\Sigma)$ denote the set of all tuples on Σ.

Definition 45

A fuzzy relation on scheme Σ is any map $R : \tau(\Sigma) \to [0,1]$. We let $F(\Sigma)$ be the set of all fuzzy relations on Σ.

If Σ and Δ are relation schemes such that $\Delta \subseteq \Sigma$, then for any tuple $t \in \tau(\Delta)$, we let t^{Σ} denote the set $\{t' \in \tau(\Sigma) \mid t'(A) = t(A), \text{ for all } A \in \Delta\}$ of all extensions of t. We extend this notion for any $T \subseteq \tau(\Delta)$ by defining $T^{\Sigma} = \cup_{t \in T} t^{\Sigma}$.

Set-theoretic operations on Fuzzy relations

Definition 46 Union:

Let R and S be fuzzy relations on scheme Σ. Then, $R \cup S$ is a fuzzy relation on scheme Σ given by

$$(R \cup S)(t) = \max\{R(t), S(t)\}, \text{ for any } t \in \tau(\Sigma).$$

Definition 47 Complement:

Let R be a fuzzy relation on scheme Σ. Then, $-R$ is a fuzzy relation on scheme Σ given by

$$(-R)(t) = 1 - R(t), \textit{for any } t \in \tau(\Sigma).$$

Definition 48 Intersection:

Let R and S be fuzzy relations on scheme Σ. Then, $R \cap S$ is a fuzzy relation on scheme Σ given by

$$(R \cap S)(t) = \min\{R(t), S(t)\}, \textit{for any } t \in \tau(\Sigma).$$

Definition 49 Difference:

Let R and S be fuzzy relations on scheme Σ. Then, $R-S$ is a fuzzy relation on scheme Σ given by

$$(R - S)(t) = \min\{R(t), 1 - S(t)\}, \textit{for any } t \in \tau(\Sigma).$$

Relation-theoretic operations on Fuzzy relations

Definition 50

Let R and S be fuzzy relations on schemes Σ and Δ, respectively. Then, the natural join (or just join) of R and S, denoted $R \bowtie S$, is a fuzzy relation on scheme Σ ∪ Δ, given by

$$(R \bowtie S)(t) = \min\{R(\pi_\Sigma(t)), S(\pi_\Delta(t))\}, \textit{for any } t \in \tau(\Sigma \cup \Delta).$$

Definition 51

Let R be a fuzzy relation on scheme Σ and let Δ ⊆ Σ. Then, the projection of R onto Δ, denoted by $\Pi_\Delta(R)$ is a fuzzy relation on scheme Δ given by

$$(\Pi_\Delta(R))(t) = \max\{R(u) | u \in t^\Sigma\}, \textit{for any } t \in \tau(\Delta).$$

Definition 52

Let R be a fuzzy relation on scheme Σ, and let F be any logic formula involving attribute names in Σ, constant symbols (denoting values in the attribute domains), equality symbol $=$, negation symbol $\neg$, and connectives $\vee$ and $\wedge$. Then, the selection of R by F, denoted $\dot{\sigma}_F(R)$, is a fuzzy relation on scheme Σ, given by

$$(\dot{\sigma}_F(R))(t) = \begin{cases} R(t) & \textit{if } t \in \sigma_F(\tau(\Sigma)) \\ 0 & \textit{Otherwise} \end{cases}$$

where σF is the usual selection of tuples satisfying F.

Neutrosophic Relations

We generalize fuzzy relations in such a manner that we are now able to assign a measure of belief and a measure of doubt to each tuple.

We shall refer to these generalized fuzzy relations as neutrosophic relations.

So, a tuple in a neutrosophic relation is assigned a measure $<\alpha, \beta>$, $0 \leq \alpha,\beta \leq 1$.

α will be referred to as the belief factor and β will be referred to as the doubt factor.

The interpretation of this measure is that we believe with confidence α and doubt with confidence β that the tuple is in the relation.

The belief and doubt confidence factors for a tuple need not add to exactly 1.

This allows for incompleteness and inconsistency to be represented.

If the belief and doubt factors add up to less than 1, we have incomplete information regarding the tuple's status in the relation and if the belief and doubt factors add up to more than 1, we have inconsistent information regarding the tuple's status in the relation.

In contrast to fuzzy relations where the grade of membership of a tuple is fixed, neutrosophic relations bound the grade of membership of a tuple to a subinterval $[\alpha, 1 - \beta]$ for the case $\alpha + \beta \leq 1$.

The operators on fuzzy relations can also be generalised for neutrosophic relations.

However, any such generalization of operators should maintain the belief system intuition behind neutrosophic relations.

This section also develops two different notions of operator generalisations.

We now formalize the notion of a neutrosophic relation.

Recall that $\tau(\Sigma)$ denotes the set of all tuples on any scheme Σ.

.

Definition 53

A neutrosophic relation R on scheme Σ is any subset of

$$\tau(\Sigma) \times [0,1] \times [0,1].$$

for any $t \in \tau(\Sigma)$, we shall denote an element of R as $<t, R(t)^+, R(t)^->$, where $R(t)^+$ is the belief factor assigned to t by R and $R(t)^-$ is the doubt factor assigned to t by R.

Let V(Σ) be the set of all neutrosophic relations on Σ.

Definition 54

A neutrosophic relation R on scheme Σ is *consistent* if $R(t)^+ + R(t)^- \leq 1$, for all $t \in \tau(\Sigma)$.

Let $C(\Sigma)$ be the set of all consistent neutrosophic relations on Σ. R is said to be *complete* if $R(t)^+ + R(t)^- \geq 1$, for all $t \in \tau(\Sigma)$.

If R is both consistent and complete, i.e. $R(t)^+ + R(t)^- = 1$, for all $t \in \tau(\Sigma)$, then it is a *total* neutrosophic relation, and let $T(\Sigma)$ be the set of all total neutrosophic relations on Σ.

Definition 55

R is said to be *pseudo-consistent* if

$$\max\{b_i | (\exists t \in \tau(\Sigma))(\exists d_i)(\langle t, b_i, d_i \rangle \in R)\} + \max\{d_i | (\exists t \in \tau(\Sigma))(\exists b_i)(\langle t, b_i, d_i \rangle \in R)\} > 1,$$

where for these $<t,b_i,d_i>$, $b_i + d_i = 1$.

Let $P(\Sigma)$ be the set of all pseudo-consistent neutrosophic relations on Σ.

Example 16

Neutrosophic relation R = {<a, 0.3, 0.7>, <a, 0.4, 0.6>, <b, 0.2, 0.5>, <c, 0.4, 0.3>} is pseudo-consistent. Because for $t = a$, max{0.3,0.4} + max{0.7,0.6} = 1.1 > 1.

It should be observed that total neutrosophic relations are essentially fuzzy relations where the uncertainty in the grade of membership is eliminated.

We make this relationship explicit by defining a one-one correspondence $\lambda\Sigma : T(\Sigma) \rightarrow F(\Sigma)$, given by $\lambda\Sigma(R)(t) = R(t)^+$, for all $t \in \tau(\Sigma)$.

This correspondence is used frequently in the following discussion.

Operator Generalisations

It is easily seen that neutrosophic relations are a generalization of fuzzy relations, in that for each fuzzy relation there is a neutrosophic relation with the same information content, but not vice versa.

It is thus natural to think of generalising the operations on fuzzy relations such as union, join, projection etc. to neutrosophic relations.

However, any such generalization should be intuitive with respect to the belief system model of neutrosophic relations.

We now construct a framework for operators on both kinds of relations and introduce two different notions of the generalization relationship among their operators.

An n-ary *operator on fuzzy relations with signature* $\langle \Sigma_1, \ldots, \Sigma_{n+1} \rangle$ is a function $\Theta : \mathcal{F}(\Sigma_1) \times \cdots \times \mathcal{F}(\Sigma_n) \to \mathcal{F}(\Sigma_{n+1})$, where $\Sigma_1, \ldots, \Sigma_{n+1}$ are any schemes. Similarly, an n-ary *operator on neutrosophic relations with signature* $\langle \Sigma_1, \ldots, \Sigma_{n+1} \rangle$ is a function $\Psi : \mathcal{V}(\Sigma_1) \times \cdots \times \mathcal{V}(\Sigma_n) \to \mathcal{V}(\Sigma_{n+1})$.

Definition 56

An operator Ψ on neutrosophic relations with signature $<\Sigma_1, ..., \Sigma_{n+1}>$ is totality preserving if for any total neutrosophic relations $R_1, ..., R_n$ on schemes $\Sigma_1, ..., \Sigma_n$, respectively, $\Psi(R_1, ..., R_n)$ is also total.

Definition 57

A totality preserving operator Ψ on neutrosophic relations with signature

$$\langle \Sigma_1, \ldots, \Sigma_{n+1} \rangle$$

is a weak generalization of an operator Θ on fuzzy relations with the same signature, if for any total neutrosophic relations $R_1, ..., R_n$ on schemes $\Sigma_1, ..., \Sigma_n$, respectively, we have

$$\lambda_{\Sigma_{n+1}}(\Psi(R_1, \ldots, R_n)) = \Theta(\lambda_{\Sigma_1}(R_1), \ldots, \lambda_{\Sigma_n}(R_n)).$$

The above definition essentially requires Ψ to coincide with Θ on total neutrosophic relations (which are in one-one correspondence with the fuzzy relations).

In general, there may be many operators on neutrosophic relations that are weak generalisations of a given operator Θ on fuzzy relations.

The behavior of the weak generalisations of Θ on even just the consistent neutrosophic relations may in general vary.

We require a stronger notion of operator generalization under which, at least when restricted to consistent intuitionistic fuzzy relations, the behavior of all the generalised operators is the same.

Before we can develop such a notion, we need that of 'representations' of a neutrosophic relation.

We associate with a consistent neutrosophic relation R the set of all (fuzzy relations corresponding to) total neutrosophic relations obtainable from R by filling in the gaps between the belief and doubt factors for each tuple.

Let the map $\mathbf{reps}_\Sigma : \mathcal{C}(\Sigma) \to 2^{\mathcal{F}(\Sigma)}$ be given by

$$\mathbf{reps}_\Sigma(R) = \{Q \in \mathcal{F}(\Sigma) \mid \bigwedge_{t_i \in \tau(\Sigma)} (R(t_i)^+ \leq Q(t_i) \leq 1 - R(t_i)^-)\}.$$

The set $\mathbf{reps}_\Sigma(R)$ contains all fuzzy relations that are 'completions' of the consistent neutrosophic relation *R*.

Observe that $\mathbf{reps}_\Sigma$ is defined only for consistent neutrosophic relations and produces sets of fuzzy relations.

Then we have following observation.

Proposition 1

For any consistent neutrosophic relation *R* on scheme Σ, repsΣ(*R*) is the singleton {λΣ(R)} iff *R* is total.

We now need to extend operators on fuzzy relations to sets of fuzzy relations.

For any operator $\Theta : \mathcal{F}(\Sigma_1) \times \cdots \times \mathcal{F}(\Sigma_n) \rightarrow \mathcal{F}(\Sigma_{n+1})$ on fuzzy relations, we let $\mathcal{S}(\Theta) : 2^{\mathcal{F}(\Sigma_1)} \times \cdots \times 2^{\mathcal{F}(\Sigma_n)} \rightarrow 2^{\mathcal{F}(\Sigma_{n+1})}$ be a map on sets of fuzzy relations defined as follows.

For any sets M_1, ..., M_n of fuzzy relations on schemes $\Sigma_1,\ldots,\Sigma_n$, respectively,

$$\mathcal{S}(\Theta)(M_1, \ldots, M_n) = \{\Theta(R_1, \ldots, R_n) \mid R_i \in M_i, \text{ for all } i, 1 \leq i \leq n\}.$$

In other words, S(Θ)(M_1, ..., M_n) is the set of Θ-images of all tuples in the cartesian product $M_1 \times \cdots \times M_n$.

We are now ready to lead up to a stronger notion of operator generalization.

Definition 58

An operator Ψ on neutrosophic relations with signature $<\Sigma_1, ..., \Sigma_{n+1}>$ is consistency preserving if for any consistent neutrosophic relations $R_1, ..., R_n$ on schemes $\Sigma_1, ..., \Sigma_n$, respectively, $\Psi(R_1, ..., R_n)$ is also consistent.

Definition 59

A consistency preserving operator Ψ on neutrosophic relations with signature $<\Sigma_1, ..., \Sigma_{n+1}>$ is a strong generalization of an operator Θ on fuzzy relations with the same signature, if for any consistent neutrosophic relations $R_1, ..., R_n$ on schemes $\Sigma_1, ..., \Sigma_n$, respectively, we have

$$\mathbf{reps}_{\Sigma_{n+1}}(\Psi(R_1, \ldots, R_n)) = \mathcal{S}(\Theta)(\mathbf{reps}_{\Sigma_1}(R_1), \ldots, \mathbf{reps}_{\Sigma_n}(R_n)).$$

Given an operator Θ on fuzzy relations, the behavior of a weak generalization of Θ is 'controlled' only over the total neutrosophic relations.

On the other hand, the behavior of a strong generalization is 'controlled' over all consistent neutrosophic relations.

This itself suggests that strong generalization is a stronger notion than weak generalization.

The following proposition makes this precise.

Proposition 2

If Ψ is a strong generalization of Θ, then Ψ is also a weak generalization of Θ.

Proof

Let <Σ_1, ..., Σ_{n+1}> be the signature of Ψ and Θ, and let R_1, ..., R_n be any total neutrosophic relations on schemes $\Sigma_1,\ldots,\Sigma_n$, respectively.

Since all total relations are consistent, and Ψ is a strong generalization of Θ, we have that

$$\mathbf{reps}_{\Sigma_{n+1}}(\Psi(R_1,\ldots,R_n)) = \mathcal{S}(\Theta)(\mathbf{reps}_{\Sigma_1}(R_1),\ldots,\mathbf{reps}_{\Sigma_n}(R_n)),$$

Proposition 1 gives us that for each i, $1 \le i \le n$, $\mathbf{reps}_{\Sigma_i}(R_i)$ is the singleton set $\{\lambda_{\Sigma_i}(R_i)\}$. Therefore, $\mathcal{S}(\Theta)(\mathbf{reps}_{\Sigma_1}(R_i),\ldots,\mathbf{reps}_{\Sigma_n}(R_n))$ is just the singleton set:

$$\{\Theta(\lambda_{\Sigma_1}(R_1),\ldots,\lambda_{\Sigma_n}(R_n))\}.$$

Here, $\Psi(R_1,\ldots,R_n)$ is total, and
$\lambda_{\Sigma_{n+1}}(\Psi(R_1,\ldots,R_n)) = \Theta(\lambda_{\Sigma_1}(R_1),\ldots,\lambda_{\Sigma_n}(R_n))$, i.e. Ψ is a weak generalization of Θ.

Though there may be many strong generalisations of an operator on fuzzy relations, they all behave the same when restricted to consistent neutrosophic relations.

In the next slides, we propose strong generalisations for the usual operators on fuzzy relations.

The proposed generalised operators on neutrosophic relations correspond to the belief system intuition behind neutrosophic relations.

First we will introduce two special operators on neutrosophic relations called split and combine to transform inconsistent neutrosophic relations into pseudo-consistent neutrosophic relations and transform pseudo-consistent neutrosophic relations into inconsistent neutrosophic relations.

Definition 60 (Split)

Let R be a neutrosophic relation on scheme Σ. Then,

$$\triangle(R) = \{\langle t,b,d\rangle | \langle t,b,d\rangle \in R \text{ and } b+d \leq 1\} \cup \{\langle t,b',d'\rangle | \langle t,b,d\rangle \in R \text{ and } b+d > 1 \text{ and } b' = b \text{ and } d' = 1-b\} \cup \{\langle t,b',d'\rangle | \langle t,b,d\rangle \in R \text{ and } b+d > 1 \text{ and } b' = 1-d \text{ and } d' = d\}.$$

It is obvious that Δ(R) is pseudo-consistent if R is inconsistent.

Definition 60 (Combine)

Let R be a neutrosophic relation on scheme Σ. Then,

$$\nabla(R) = \{\langle t,b',d'\rangle | (\exists b)(\exists d)((\langle t,b',d\rangle \in R \text{ and } (\forall b_i,d_i)(\langle t,b_i,d_i\rangle \rightarrow b' \geq b_i) \text{ and } \langle t,b,d'\rangle \in R \text{ and } (\forall b_i)(\forall d_i)(\langle t,b_i,d_i\rangle \rightarrow d' \geq d_i))\}.$$

It is obvious that $\nabla(R)$ is inconsistent if R is pseudo-consistent.

Note that strong generalization defined above only holds for consistent or pseudo-consistent neutrosophic relations.

For any arbitrary paraconsisent intuitionistic fuzzy relations, we should first use split operation to transform them into non inconsistent neutrosophic relations and apply the set-theoretic and relation-theoretic operations on them and finally use combine operation to transform the result into arbitrary neutrosophic relation.

For the simplification of notation, the following generalized algebra is defined under such assumption.

Generalized Algebra on Neutrosophic Relations

We present one strong generalization each for the fuzzy relation operators such as union, join, projection.

To reflect generalization, a hat is placed over a fuzzy relation operator to obtain the corresponding neutrosophic relation operator.

For example, $\bowtie$ denotes the natural join among fuzzy relations, and $\bowtie$ denotes natural join on neutrosophic relations.

These generalized operators maintain the belief system intuition behind neutrosophic relations.

Set-Theoretic Operators

We first generalize the two fundamental set-theoretic operators, union and complement.

Definition 62

Let R and S be neutrosophic relations on scheme Σ. Then,

(a) *the* union *of R and S, denoted $R \mathbin{\widehat{\cup}} S$, is a neutrosophic relation on scheme Σ, given by*

$$(R \mathbin{\widehat{\cup}} S)(t) = \langle \max\{R(t)^+, S(t)^+\}, \min\{R(t)^-, S(t)^-\}\rangle, \text{ for any } t \in \tau(\Sigma);$$

(b) *the* complement *of R, denoted $\widehat{-}\, R$, is a neutrosophic relation on scheme Σ, given by*

$$(\widehat{-}\, R)(t) = \langle R(t)^-, R(t)^+\rangle, \text{ for any } t \in \tau(\Sigma).$$

An intuitive appreciation of the union operator can be obtained as follows: Given a tuple t, since we believed that it is present in the relation R with confidence $R(t)^+$ and that it is present in the relation S with confidence $S(t)^+$, we can now believe that the tuple t is present in the "either-R-or-S" relation with confidence which is equal to the larger of $R(t)^+$ and $S(t)^+$.

Using the same logic, we can now believe in the absence of the tuple t from the "either-R-or-S" relation with confidence which is equal to the smaller (because t must be absent from both R and S for it to be absent from the union) of $R(t)^-$ and $S(t)^-$.

The definition of complement and of all the other operators on neutrosophic relations defined later can (and should) be understood in the same way.

An intuitive appreciation of the union operator can be obtained as follows: Given a tuple t, since we believed that it is present in the relation R with confidence $R(t)^+$ and that it is present in the relation S with confidence $S(t)^+$, we can now believe that the tuple t is present in the "either-R-or-S" relation with confidence which is equal to the larger of $R(t)^+$ and $S(t)^+$.

Using the same logic, we can now believe in the absence of the tuple t from the "either-R-or-S" relation with confidence which is equal to the smaller (because t must be absent from both R and S for it to be absent from the union) of $R(t)^-$ and $S(t)^-$.

The definition of complement and of all the other operators on neutrosophic relations defined later can (and should) be understood in the same way.

Proposition 3

The operators $\widehat{\cup}$ and unary $\widehat{-}$ on neutrosophic relations are strong generalisations of the operators $\cup$ and unary $-$ on fuzzy relations.

Definition 63

Let R and S be neutrosophic relations on scheme Σ. Then

(a) *the* intersection *of R and S, denoted $R \mathbin{\widehat{\cap}} S$, is a neutrosophic relation on scheme Σ, given by*

$$(R \mathbin{\widehat{\cap}} S)(t) = \langle \min\{R(t)^+, S(t)^+\}, \max\{R(t)^-, S(t)^-\}\rangle, \ \textit{for any } t \in \tau(\Sigma);$$

(b) *the* difference *of R and S, denoted $R \mathbin{\widehat{-}} S$, is a neutrosophic relation on scheme Σ, given by*

$$(R \mathbin{\widehat{-}} S)(t) = \langle \min\{R(t)^+, S(t)^-\}, \max\{R(t)^-, S(t)^+\}\rangle, \ \textit{for any } t \in \tau(\Sigma);$$

The following proposition relates the intersection and difference operators in terms of the more fundamental set-theoretic operators union and complement.

Proposition 4

For any neutrosophic relations R and S on the same scheme, we have

$$\begin{aligned} R \mathbin{\widehat{\cap}} S &= \widehat{-}(\widehat{-}R \mathbin{\widehat{\cup}} \widehat{-}S), \\ R \mathbin{\widehat{-}} S &= \widehat{-}(\widehat{-}R \mathbin{\widehat{\cup}} S). \end{aligned}$$

Relation-Theoretic Operators

We now define some relation-theoretic algebraic operators on neutrosophic relations.

Definition 64

Let R and S be neutrosophic relations on schemes Σ and Δ, respectively.

Then, the natural join (further for short called join) of R and S, denoted $R \hat{\bowtie} S$, is a neutrosophic relation on scheme Σ∪Δ, given by

$$(R \hat{\bowtie} S)(t) = \langle \min\{R(\pi_\Sigma(t))^+, S(\pi_\Delta(t))^+\}, \max\{R(\pi_\Sigma(t))^-, S(\pi_\Delta(t))^-\}\rangle,$$

where π is the usual projection of a tuple.

It is instructive to observe that, similar to the intersection operator, the minimum of the belief factors and the maximum of the doubt factors are used in the definition of the join operation.

Proposition 5

$\bowtie$ is a strong generalization of $\hat{\bowtie}$.

Definition 65

Let R be a neutrosophic relation on scheme Σ, and $\Delta \subseteq \Sigma$.

Then, the projection of R onto Δ, denoted $\widehat{\pi}_\Delta(R)$, is a neutrosophic relation on scheme Δ, given by

$$(\widehat{\pi}_\Delta(R))(t) = \langle \max\{R(u)^+ | u \in t^\Sigma\}, \min\{R(u)^- | u \in t^\Sigma\} \rangle.$$

The belief factor of a tuple in the projection is the maximum of the belief factors of all of the tuple's extensions onto the scheme of the input neutrosophic relation.

Moreover, the doubt factor of a tuple in the projection is the minimum of the doubt factors of all of the tuple's extensions onto the scheme of the input neutrosophic relation.

We present the selection operator next.

Definition 66

Let R be a neutrosophic relation on scheme Σ, and let F be any logic formula involving attribute names in Σ, constant symbols (denoting values in the attribute domains), equality symbol $=$, negation symbol $\neg$, and connectives $\vee$ and $\wedge$.

Then, the selection of R by F, denoted $\hat{\sigma}_F(R)$, is a neutrosophic relation on scheme Σ, given by

$$(\widehat{\sigma}_F(R))(t) = \langle \alpha, \beta \rangle, \ \textit{where}$$

$$\alpha = \begin{cases} R(t)^+ & \textit{if } t \in \sigma_F(\tau(\Sigma)) \\ 0 & \textit{otherwise} \end{cases} \qquad \textit{and} \qquad \beta = \begin{cases} R(t)^- & \textit{if } t \in \sigma_F(\tau(\Sigma)) \\ 1 & \textit{otherwise} \end{cases}$$

where σ_F is the usual selection of tuples satisfying F from ordinary relations.

If a tuple satisfies the selection criterion, it's belief and doubt factors are the same in the selection as in the input neutrosophic relation.

In the case where the tuple does not satisfy the selection criterion, its belief factor is set to 0 and the doubt factor is set to 1 in the selection.

Proposition 6

The operators $\hat{\pi}$ and $\hat{\sigma}$ are strong generalisations of π and σ, respectively.

Example 17

Relation schemes are sets of attribute names, but in this example we treat them as ordered sequences of attribute names (which can be obtained through permutation of attribute names), so tuples can be viewed as the usual lists of values.

Let $\{a, b, c\}$ be a common domain for all attribute names, and let R and S be the following neutrosophic relations on schemes $<X, Y>$ and $<Y, Z>$ respectively.

t	$R(t)$
(a, a)	$\langle 0, 1\rangle$
(a, b)	$\langle 0, 1\rangle$
(a, c)	$\langle 0, 1\rangle$
(b, b)	$\langle 1, 0\rangle$
(b, c)	$\langle 1, 0\rangle$
(c, b)	$\langle 1, 1\rangle$

t	$S(t)$
(a, c)	$\langle 1, 0\rangle$
(b, a)	$\langle 1, 1\rangle$
(c, b)	$\langle 0, 1\rangle$

For other tuples which are not in the neutrosophic relations $R(t)$ and $S(t)$, their $<\alpha, \beta> = <0, 0>$ which means no any information available.

Because R and S are inconsistent, we first use split operation to transform them into pseudo-consistent and apply the relation-theoretic operations on them and transform the result back to arbitrary neutrosophic set using combine operation.

T_1, T_2 and T_3 are shown below:

t	$T_1(t)$
(a,a,a)	$\langle 0,1\rangle$
(a,a,b)	$\langle 0,1\rangle$
(a,a,c)	$\langle 0,1\rangle$
(a,b,a)	$\langle 0,1\rangle$
(a,b,b)	$\langle 0,1\rangle$
(a,b,c)	$\langle 0,1\rangle$
(a,c,a)	$\langle 0,1\rangle$
(a,c,b)	$\langle 0,1\rangle$
(a,c,c)	$\langle 0,1\rangle$
(b,b,a)	$\langle 1,1\rangle$
(b,c,b)	$\langle 0,1\rangle$
(c,b,a)	$\langle 1,1\rangle$
(c,b,b)	$\langle 0,1\rangle$
(c,b,c)	$\langle 0,1\rangle$
(c,c,b)	$\langle 0,1\rangle$

t	$T_2(t)$
(a,a)	$\langle 0,1\rangle$
(a,b)	$\langle 0,1\rangle$
(a,c)	$\langle 0,1\rangle$
(b,a)	$\langle 1,0\rangle$
(c,a)	$\langle 1,0\rangle$

t	$T_3(t)$
(a,a)	$\langle 0,1\rangle$
(a,b)	$\langle 0,1\rangle$
(a,c)	$\langle 0,1\rangle$
(b,a)	$\langle 1,0\rangle$
(b,b)	$\langle 0,1\rangle$
(c,a)	$\langle 1,0\rangle$
(c,c)	$\langle 0,1\rangle$

An Application

Consider the target recognition example presented in

V. S. Subrahmanian, *Amalgamating knowledge bases*, ACM Transactions on Database Systems **19** (1994), no. 2, 291–331.

Here, an autonomous vehicle needs to identify objects in a hostile environment such as a military battlefield.

The autonomous vehicle is equipped with a number of sensors which are used to collect data, such as speed and size of the objects (tanks) in the battlefield.

Associated with each sensor, we have a set of rules that describe the type of the object based on the properties detected by the sensor.

Let us assume that the autonomous vehicle is equipped with three sensors resulting in data collected about radar readings, of the tanks, their gun characteristics and their speeds.

What follows is a set of rules that associate the type of object with various observations.

Radar Readings:

- Reading r_1 indicates that the object is a T-72 tank with belief factor 0.80 and doubt factor 0.15.
- Reading r_2 indicates that the object is a T-60 tank with belief factor 0.70 and doubt factor 0.20.
- Reading r_3 indicates that the object is not a T-72 tank with belief factor 0.95 and doubt factor 0.05.
- Reading r_4 indicates that the object is a T-80 tank with belief factor 0.85 and doubt factor 0.10.

Gun Characteristics:

- Characteristic c_1 indicates that the object is a T-60 tank with belief factor 0.80 and doubt factor 0.20.
- Characteristic c_2 indicates that the object is not a T-80 tank with belief factor 0.90 and doubt factor 0.05.
- Characteristic c_3 indicates that the object is a T-72 tank with belief factor 0.85 and doubt factor 0.10.

Speed Characteristics:

- Low speed indicates that the object is a T-60 tank with belief factor 0.80 and doubt factor 0.15.
- High speed indicates that the object is not a T-72 tank with belief factor 0.85 and doubt factor 0.15.
- High speed indicates that the object is not a T-80 tank with belief factor 0.95 and doubt factor 0.05.
- Medium speed indicates that the object is not a T-80 tank with belief factor 0.80 and doubt factor 0.10.

These rules can be captured in the following three neutrosophic relations:

Radar Rules

Reading	Object	Confidence Factors
r_1	T-72	$\langle 0.80, 0.15 \rangle$
r_2	T-60	$\langle 0.70, 0.20 \rangle$
r_3	T-72	$\langle 0.05, 0.95 \rangle$
r_4	T-80	$\langle 0.85, 0.10 \rangle$

Gun Rules

Reading	Object	Confidence Factors
c_1	T-60	$\langle 0.80, 0.20 \rangle$
c_2	T-80	$\langle 0.05, 0.90 \rangle$
c_3	T-72	$\langle 0.85, 0.10 \rangle$

Speed Rules

Reading	Object	Confidence Factors
low	T-60	$\langle 0.80, 0.15 \rangle$
high	T-72	$\langle 0.15, 0.85 \rangle$
high	T-80	$\langle 0.05, 0.95 \rangle$
medium	T-80	$\langle 0.10, 0.80 \rangle$

The autonomous vehicle uses the sensors to make observations about the different objects and then uses the rules to determine the type of each object in the battlefield.

It is quite possible that two different sensors may identify the same object as of different types, thereby introducing inconsistencies.

Let us now consider three objects o1, o2 and o3 which need to be identified by the autonomous vehicle.

Let us assume the following observations made by the three sensors about the three objects.

Once again, we assume certainty factors (maybe derived from the accuracy of the sensors) are associated with each observation.

Radar Data

Object-id	Reading	Confidence Factors
o_1	r_3	$\langle 1.00, 0.00 \rangle$
o_2	r_1	$\langle 1.00, 0.00 \rangle$
o_3	r_4	$\langle 1.00, 0.00 \rangle$

Gun Data

Object-id	Reading	Confidence Factors
o_1	c_3	$\langle 0.80, 0.10 \rangle$
o_2	c_1	$\langle 0.90, 0.10 \rangle$
o_3	c_2	$\langle 0.90, 0.10 \rangle$

Speed Data

Object-id	Reading	Confidence Factors
o_1	high	$\langle 0.90, 0.10 \rangle$
o_2	low	$\langle 0.95, 0.05 \rangle$
o_3	medium	$\langle 0.80, 0.20 \rangle$

Given these observations and the rules, we can use the following algebraic expression to identify the three objects:

$$\hat{\pi}_{\text{Object-id,Object}}(\text{Radar Data} \hat{\bowtie} \text{Radar Rules}) \hat{\cap}$$
$$\hat{\pi}_{\text{Object-id,Object}}(\text{Gun Data} \hat{\bowtie} \text{Gun Rules}) \hat{\cap}$$
$$\hat{\pi}_{\text{Object-id,Object}}(\text{Speed Data} \hat{\bowtie} \text{Speed Rules})$$

I_1	q_1	$\langle 0.9, 0.2\rangle$
I_1	q_2	$\langle 1.0, 0.0\rangle$
I_1	q_3	$\langle 0.1, 0.8\rangle$
I_2	q_1	$\langle 1.0, 1.0\rangle$
I_2	q_3	$\langle 0.8, 0.3\rangle$

Table EVAL

The intuition behind the intersection is that we would like to capture the common (intersecting) information among the three sensor data.

Evaluating this expression, we get the following neutrosophic relation:

Object-id	Object	Confidence Factors
o_1	T-72	$\langle 0.05, 0.0\rangle$
o_2	T-80	$\langle 0.0, 0.05\rangle$
o_3	T-80	$\langle 0.05, 0.0\rangle$

It is clear from the result that by the given information, we could not infer any useful information that is we could not decide the status of objects o_1, o_2 and o_3.

Conclusions

We have presented a generalization of fuzzy relations, intuitionistic fuzzy relations (interval-valued fuzzy relations) and paraconsistent relations, called neutrosophic relations, in which we allow the representation of confidence (belief and doubt) factors with each tuple.

The algebra on fuzzy relations is appropriately generalized to manipulate neutrosophic relations.

Various possibilities exist for further study in this area. Recently, there has been some work in extending logic programs to involve quantitative paraconsistency.

Paraconsistent logic programs allow negative atoms to appear in the head of clauses (thereby resulting in the possibility of dealing with inconsistency), and probabilistic logic programs associate confidence measures with literals and with entire clauses.

The semantics of these extensions of logic programs have already been presented, but implementation strategies to answer queries have not been discussed.

We propose to use the model introduced in this chapter in computing the semantics of these extensions of logic programs.

Exploring application areas is another important thrust of our research.

We developed two notions of generalising operators on fuzzy relations for neutrosophic relations.

Of these, the stronger notion guarantees that any generalised operator is “well-behaved” for neutrosophic relation operands that contain consistent information.

For some well-known operators on fuzzy relations, such as union, join, projection, we introduced generalised operators on neutrosophic relations.

These generalised operators maintain the belief system intuition behind neutrosophic relations, and are shown to be “well-behaved” in the sense mentioned above.

Our data model can be used to represent relational information that may be incomplete and inconsistent.

As usual, the algebraic operators can be used to construct queries to any database systems for retrieving vague information.

5. Soft Semantic Web Services Agent

Introduction

Web services technology is critical for the success of business integration and other application fields such as bioinformatics.

However, there are two challenges facing the practicality of Web services:

(a) efficient location of the Web service registries that contain the requested Web services;

(b) efficient retrieval of the requested services from these registries with high quality of service (QoS).

The main reason for this problem is that current Web services technology is not semantically oriented.

Several proposals have been made to add semantics to Web services to facilitate discovery and composition of relevant Web services.

Such proposals are being referred to as Semantic Web services (SWS).

However, most of these proposals do not address the second problem of retrieval of Web services with high QoS.

We propose a framework called Soft Semantic Web Services Agent (soft SWS agent) for providing high QoS Semantic Web services using soft computing methodology.

Since different application domains have different requirements for QoS, it is impractical to use classical mathematical modeling methods to evaluate the QoS of semantic Web services.

We use neutrosophic neural networks with Genetic Algorithms (GA) as our study case.

Simulation results show that the soft SWS agent methodology is extensible and scalable to handle fuzzy, uncertain and inconsistent QoS metrics effectively.

We propose a framework called soft semantic Web services agent (soft SWS agent) to provide high QoS semantic Web services based on specific domain ontology such as gnome.

The soft SWS agent could solve the forementioned two challenges effectively and efficiently.

The soft SWS agent itself is implemented as a semantic Web service and comprises of six components:

(a) Registries Crawler,
(b) Repository,
(c) Inquiry Server,
(d) Publish Server,
(e) Agent Communication Server,
(f) Intelligent Inference Engine.

The core of the soft SWS agent is Intelligent Inference Engine (IIE). It uses soft computing technologies to evaluate the entire QoS of semantic Web services using both functional and non-functional properties.

We use semantic Web services for bioinformatics as a case study. We employ neutrosophic neural networks with Genetic Algorithms (GA) for the IIE component of our soft SWS agent.

The case study illustrates the flexibility and reliability of soft computing methodology for handling fuzzy and uncertain linguistic information.

For example, capability of a Web service is fuzzy. It is unreasonable to use crisp values to describe it. So we can use several linguistic variables such as a ”little bit low” and ”a little bit high” to express the capability of services.

Background

Traditional Web services

Web services are modular, self-describing, and self-contained applications that are accessible over the internet.

The core components of the Web services infrastructure are XML based standards like SOAP, WSDL, and UDDI.

SOAP is the standard messaging protocol for Web services. SOAP messages consist of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. WSDL is an XML format to describe Web services as collections of communication endpoints that can exchange certain messages.

A complete WSDL service description provides two pieces of information: an application-level service description (or abstract interface), and the specific protocol-dependent details that users must follow to access the service at a specified concrete service endpoint.

The UDDI specifications offer users a unified and systematic way to find service providers through a centralized registry of services that is roughly equivalent to an automated online "phone directory" of Web services.

UDDI provides two basic specifications that define a service registry's structure and operation.

One is a definition of the information to provide about each service and how to encode it and the other is a publish and query API for the registry that describes how this information can be published and accessed.

Semantic Web

The current Web is just a collection of documents which are human readable but not machine processable.

In order to remedy this disadvantage, the concept of semantic Web is proposed to add semantics to the Web to facilitate the information finding, extracting, representing, interpreting and maintaining.

> "The semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

T. Berners-Lee, J. Hendler, and O. Lassila, *The semantic web*, Scientific American **284** (2001), no. 5, 34–43.

The core concept of semantic Web is ontology.

> "Ontology is a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic."

J. Hendler, *Agents and the semantic web*, IEEE Intelligent Systems **16** (2001), no. 2, 30–37.

There are many semantic Web technologies available today, such as RDF [rdfb], RDFS [rdfa], DAML+OIL [damb] and OWL [owlb].

The description logics are used as the inference mechanism for current semantic Web technologies.

There are some drawbacks in the description logics. It cannot handle fuzziness and uncertainty associated with concept membership.

The current research trend is to combine soft computing with semantic Web.

Semantic Web Services

The industry is proposing Web services to transform the Web from “passive state”–repository of static documents to “positive state”–repository of dynamic services.

Unfortunately, the current Web services standards are not semantic-oriented.

They are awkward for service discovery, invocation, composition, and monitoring.

So it is natural to combine the semantic Web with Web services, the so-called semantic Web services.

Several projects have been initiated to design the framework for semantic Web services such as OWL-S, IRS-II, WSMF and METEOR-S.

For example, OWL-S 1.0 which is based on OWL is the upper ontology for services.

OWL-S 1.0 has three subontologies: ServiceProfile, ServiceModel and ServiceGrounding.

The service profile tells "what the service does"; this is, it gives the types of information needed by a service-seeking agent to determine whether the service meets its needs.

The service model tells "how the service works"; that is, it describes what happens when the service is carried out.

A service grounding specifies the details of how an agent can access a service.

Typically a grounding will specify a communication protocol, message formats, and other service-specific details such as port numbers used in contacting the service.

In addition, the grounding must specify, for each abstract type specified in the ServiceModel, an unambiguous way of exchanging data elements of that type with the service.

Soft Computing Methodology

> "Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximations."

L. Zadeh, *Fuzzy logic, neural networks, and soft computing*, Communications of the ACM **37** (1994), 77–84.

The principal constituents of soft computing are fuzzy logic, neural networks, and generic algorithms.

More and more technologies will join into the soft computing framework in the near future.

Fuzzy logic is primarily concerned with handling imprecision and uncertainty, neural computing focuses on simulating human being's learning process, and genetic algorithms simulate the natural selection and evolutionary processes to perform randomized global search.
Each component of soft computing is complementary to each other.

Using combinations of several technologies such as fuzzy-neural systems will generally get better solutions.

QoS Model

Different applications generally have different requirements of QoS dimensions.

Rommel,

G. Rommel, *Simplicity wins: how germany's mid-sized industrial companies succeed*, Harvard Business School Press, Boston, Mass, 1995.

Stalk and Hout

G. Stalk and T. Hout, *Competing against time: how timebased competition is reshaping global markets*, Free Press, New York, 1990.

investigate the features with which successful companies assert themselves in the competitive world markets.

Their result showed that success is based on three essential dimensions: time, cost and quality.

For middleware systems, Frlund and Koisinen

S. Frlund and J. Koisinen, *Quality-of-service specification in distributed object systems*, Distributed System Engineering Journal **5** (1998), no. 4, 179–202.

present a set of practical dimensions for distributed object systems reliability and performance, which include TTR (time to repair), TTF (time to failure), availability, failure masking, and server failure.

Gardaso, Miller, Sheth and Arnold

J. Gardoso, J. Miller, A. Sheth, and J. Arnold, *Modelling quality of service for workflows and web service processes.*

propose a QoS model for workflows and Web services processes based on four dimensions: time, cost, reliability and fidelity.

We construct a QoS model for semantic Web services.

It is composed of the following dimensions:

capability,
response time,
trustworthiness.

In order to be more precise, we give our definitions of the three dimensions as follows:

1. The capability of a semantic Web service can be defined as the degree to which its functional properties match with the required functional properties of the semantic Web service requestor;
2. The response time of a semantic Web service represents the time that elapses between service requests arrival and the completion of that service request. Response time is the sum of waiting time and actual processing time;
3. The trustworthiness of a semantic Web services is the extent to which it is consistent, reliabile, competent, and honest.

Architecture of Extensible Soft SWS Agent

The extensible soft SWS agent can provide high QoS semantic Web services based on specific ontology.

The extensible SWS agent uses centralized client/server architecture internally.

But itself can also be and should be implemented as a semantic Web service based on specific service ontology.

The extensible soft SWS agent comprises of six components:

(a) Registries Crawler,
(b) Repository,
(c) Inquiry Server,
(d) Publish Server,
(e) Agent Communication Server,
(f) Intelligent Inference Engine.

The high level architecture of the extensible soft SWS agent is shown in *Figure* below. Each of the components is described next.

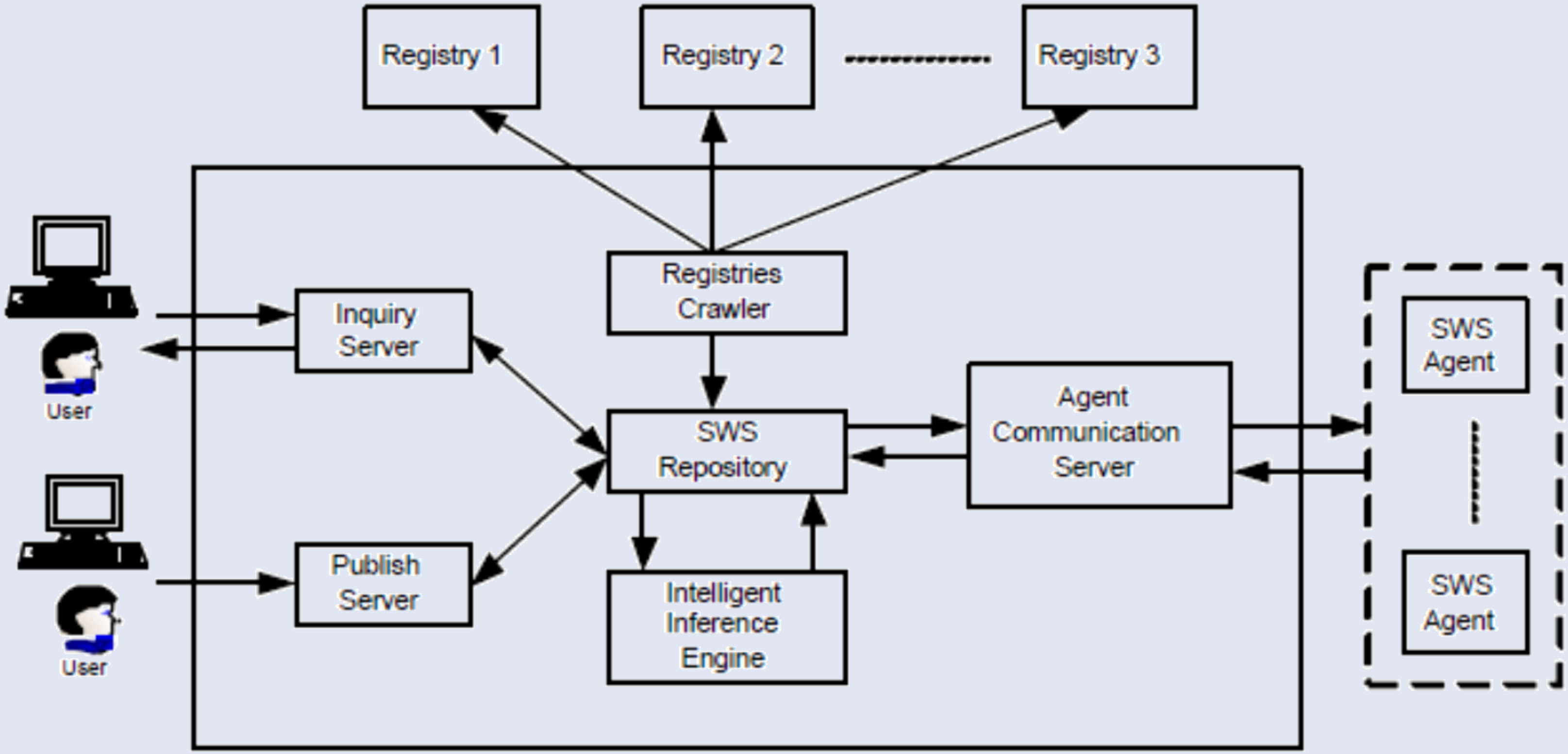


Figure 6: Architecture of the Extensible Soft SWS Agent

Registries Crawler

The current UDDI registry only supports keyword based search for the Web services description.

Under the Semantic Web environment, UDDI registry must be extended to be ontology compatible which supports semantic matching of semantic Web services' capabilities.

One possible way is to map the OWL-S service profiles into current UDDI registry's data structure.

Semantic Web service providers will publish the service profiles of semantic Web services in the public or private specific service ontology-oriented UDDI registries or directly on their semantic Web sites.

The specific ontology based semantic Web services registries crawler has two tasks:

1. Accessing these public and private specific service ontology-oriented UDDI registries using UDDI query API to fetch the service profiles, transforming them into the format supported by our repository, and storing them into the repository using the publish API of our repository;

2. Crawling the semantic Web sites hosting the specific ontology based semantic Web services directly to get the service profiles, transforming them into the format supported by the repository, and storing them into repository using the publish API for the repository.

The registries crawler should be multithreaded and should be available 24x7. The registries crawler must also be provided the information of highest level specific service ontology before its execution.

SWS Repository

The specific ontology based semantic Web services repository will store service profiles of semantic Web services.

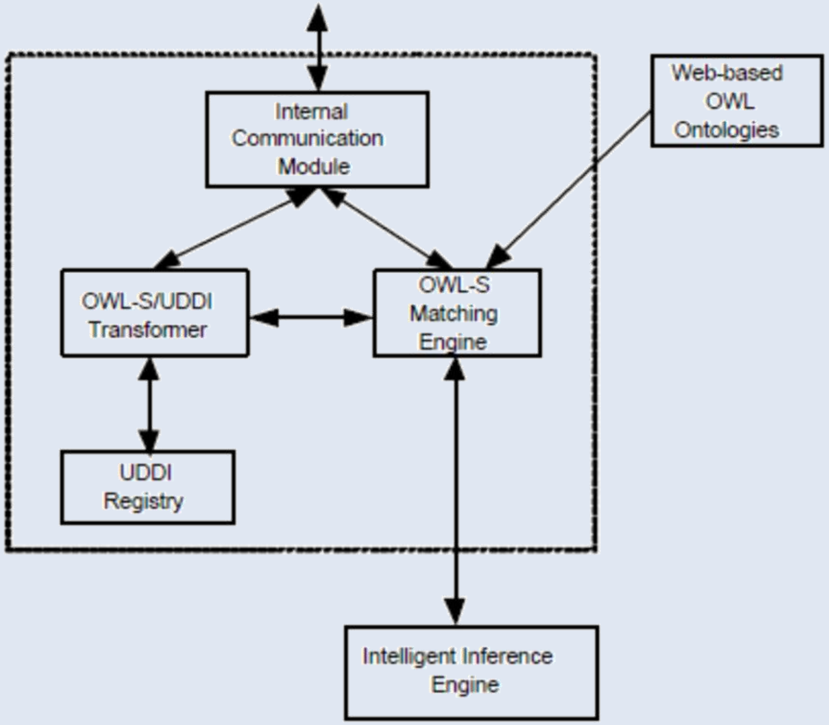


Figure 7: Architecture of Repository

The internal communication module provides the communication interface between the repository and the registries crawler, inquiry server, publish server, and the agent communication server.

If a message is an advertisement, the internal communication module sends it to the OWL-S/UDDI transformer that constructs a UDDI service description using information about the service provider and the service name.

The result of publishing with the UDDI is a reference ID of the service.

This ID combined with the capability description and non-functional properties of the advertisement are sent to the OWL-S matching engine that stores the advertisement for capability matching.

If a message is a query, the internal communication module sends the request to the OWL-S matching engine that performs the capability matching.

After calculating the degree of capability, the OWL-S matching engine will feed the degree of capability and non-functional properties to the intelligent inference engine to get the entire Quality of Servie (QoS).

The service with highest QoS will be selected.

The result of the selection is the advertisement of the providers selected and a reference to the UDDI service record.

The combination of UDDI records and advertisements is then sent to the inquiry server.

If the required service does not exist, OWL-S matching engine will transfer the query to the agent communication server through the internal communication module.

The matching algorithm used by OWL-S matching engine is based on the modified algorithm described in:

M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, *Semantic matching of web services capabilities*, Proceedings of ISWC 2002, 2002.

The modified algorithm considers not only the inputs, outputs, preconditions and effects, but also service name.

Inquiry Server

The specific ontology based semantic Web services inquiry server provides two kinds of query interface: a programmatic API to other semantic Web services or agents and a Web-based interface for the human user.

Both interfaces support keyword oriented query as well as capability oriented searches.

For capability oriented query, the inquiry server transforms the service request profile into the format supported by the repository such as OWL-S service profile and sends the query message to the internal communication module of the repository.

The internal communication module sends the service profile to the OWL-S matching engine and returns back the requested advertisement to the inquiry server and then on to the service requestor.

The process is shown in Figure, next slide.

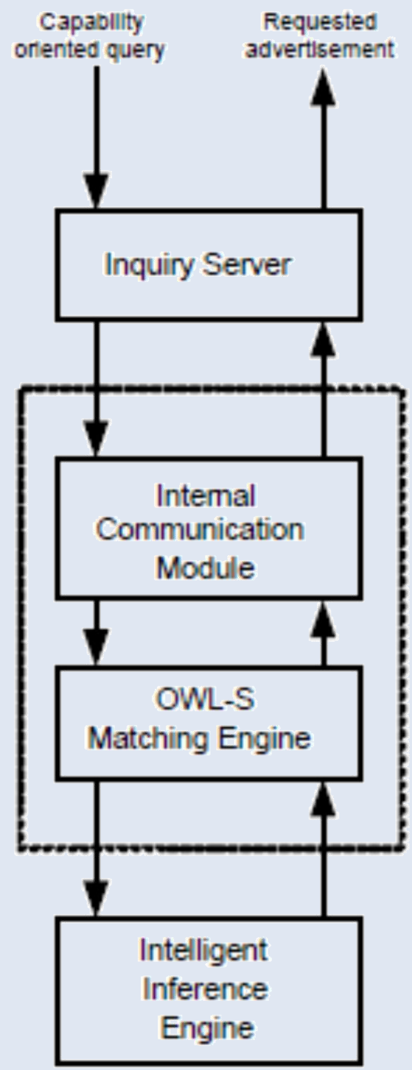


Figure 8: Capability oriented query.

For the keyword oriented queries, the inquiry server will directly send the query string to the internal communication module as a query message and the internal communication module sends the query string to the UDDI Registry and returns back the requested UDDI records to the inquiry server and then on to the service requestor.

The process is shown in next Figure:

We use SOAP as a communication protocol between service requestors and the inquiry server.

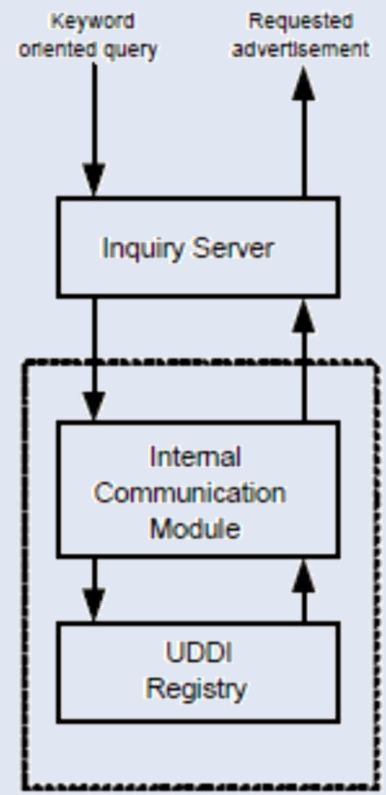


Figure 9: Keyword oriented query.

Publish Server

The specific ontology based semantic Web services publish server provides the publishing service for other agents and human users.

It has two kinds of interface.

One is the programmatic API to other semantic Web services or agents and another is for the human user which is Web-based.

The publish server will transform the service advertisement into the format supported by the repository such as OWL-S service profile and sends the publish message to the internal communication module.

The internal communication module sends the transformed OWL-S service profile to the OWL-S/UDDI transformer. The OWL-S/UDDI transformer will map the OWL-S service profile into UDDI registries data structure, and store the OWL-S service profile and reference ID of service into OWL-S matching engine.

If the advertised semantic Web services are not in the domain of the soft SWS agent, the internal communication server will transfer the advertisements to the agent communication server which will try to publish the advertisements into other soft SWS agents.

SOAP is used as a communication protocol between service publisher and the publisher server.

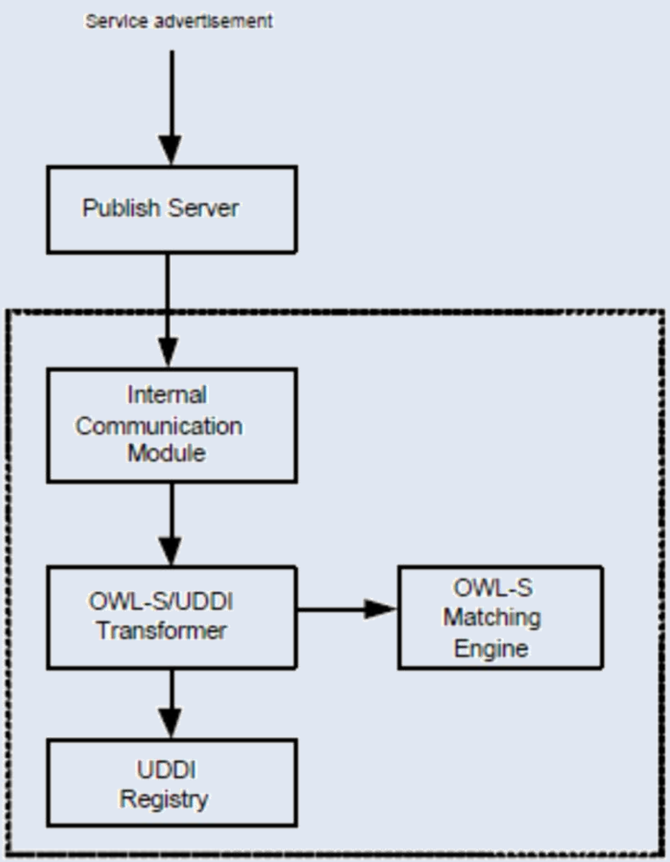


Figure 10: Publish service advertisement.

Agent Communication Server

The soft semantic Web services agent communication server uses a certain communication protocol such as Knowledge Query and Manipulation Language (KQML) and Agent Communication Language (ACL) to communicate with other soft SWS agents.

If the current soft SWS agent cannot fulfill the required services (query and publish), the agent communication server is responsible for transferring the requirements to other soft SWS agents, getting results back, and conveying the results back to the service requestors.

The current KQML and ACL should be extended to be ontology-compatible to facilitate the semantic oriented communication.

Intelligent Inference Engine

The intelligent inference engine (IIE) is the core of the soft SWS agent.

The soft SWS agent is extensible because IIE uses soft computing methodology to calculate the QoS of the semantic Web services with multidimensional QoS metrics.

IIE gets the degree of capability matching and non-functional properties' values from OWL-S matching engine and returns back the whole QoS to OWL-S matching engine.

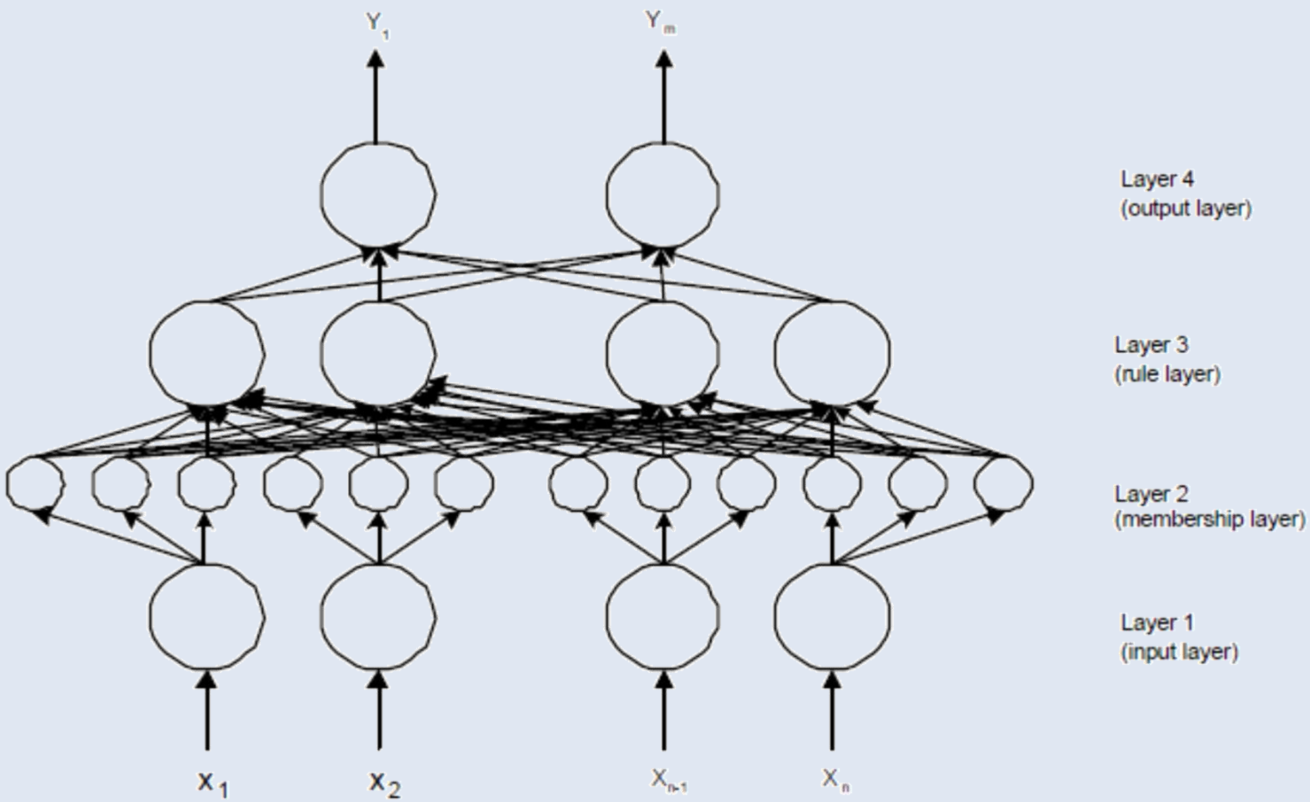


Figure: 11 Schematic diagram of Neutrosophic Neural Network.

Design of Intelligent Inference Engine

A schematic diagram of the four-layered neutrosophic neural network was shown in previous *Figure*.

Nodes in layer one are input nodes representing input linguistic variables.

Nodes in layer two are membership nodes.

Membership nodes are truth-membership node, indeterminacy-membership node and falsity-membership node, which are responsible for mapping an input linguistic variable into three possibility distributions for that variable.

The rule nodes reside in layer three.

The last layer contains the output variable nodes.

The metrics of QoS of Semantic Web services are multidimensional.

For illustration of specific ontology based Semantic Web services for bioinformatics, we decide to use capability, response time and trustworthiness as our inputs and whole QoS as output.

The neutrosophic logic system is based on TSK model.

Input neutrosophic sets

Let x represent capability, y represent response time and z represent trustworthiness.

We scale the capability, response time and trustworthiness to [0,10] respectively.

Neutrosophic rule bases

We design the neutrosophic rule base based on the TSK model.

A neutrosophic rule is shown below:

$$\text{IF } x \text{ is } I_1 \text{ and } y \text{ is } I_2 \text{ and } z \text{ is } I_3 \text{ THEN } O \text{ is } a_{i,1} * x + a_{i,2} * y + a_{i,3} * Z + a_{i,4}.$$

where, I_1, I_2 and I_3 are in low, middle, and high respectively and i in [1,27].

There are totally 27 neutrosophic rules.

The $a_{i,j}$ are consequent parameters which will be obtained by training phase of neutrosophic neural network using genetic algorithm.

Design of deneutrosophication

Suppose, for certain inputs x, y and z, there are m fired neutrosophic rules.

To calculate the firing strength of j^{th} rule, we use the formula:

$$W^j = W^j_x * W^j_y * W^j_z, \tag{4.1}$$

where

$$W^j_x = (0.5 * t_x(x) + 0.35 * (1 - f_x(x)) + 0.025 * i_x(x) + 0.05),$$
$$W^j_y = (0.5 * t_y(y) + 0.35 * (1 - f_y(y)) + 0.025 * i_y(y) + 0.05),$$
$$W^j_z = (0.5 * t_z(z) + 0.35 * (1 - f_z(z)) + 0.025 * i_z(z) + 0.05),$$

where t_x, f_x, i_x, t_y, f_y, i_y, t_z, f_z, i_z are the truth-membership, falsity-membership, indeterminacy-membership of neutrosophic inputs x, y, z, respectively.

So the crisp output is:

$$O = \sum_{j=1}^{m} W^j * (a_{j,1} * x + a_{j,2} * y + a_{j,3} * z + a_{j,4}) / (\sum_{j=1}^{m} W^j) \tag{4.2}$$

Genetic algorithms

GA is a model of machine learning which derives its behavior form a metaphor of the processes of evolution in nature.

This is done by creation within a machine of a population of individuals represented by chromosomes.

Here we use real-coded scheme.

Given the range of parameters (coefficients of linear equations in TSK model), the system uses the derivate-free random search-GA to learn to find the near optimal solution by the fitness function through the training data.

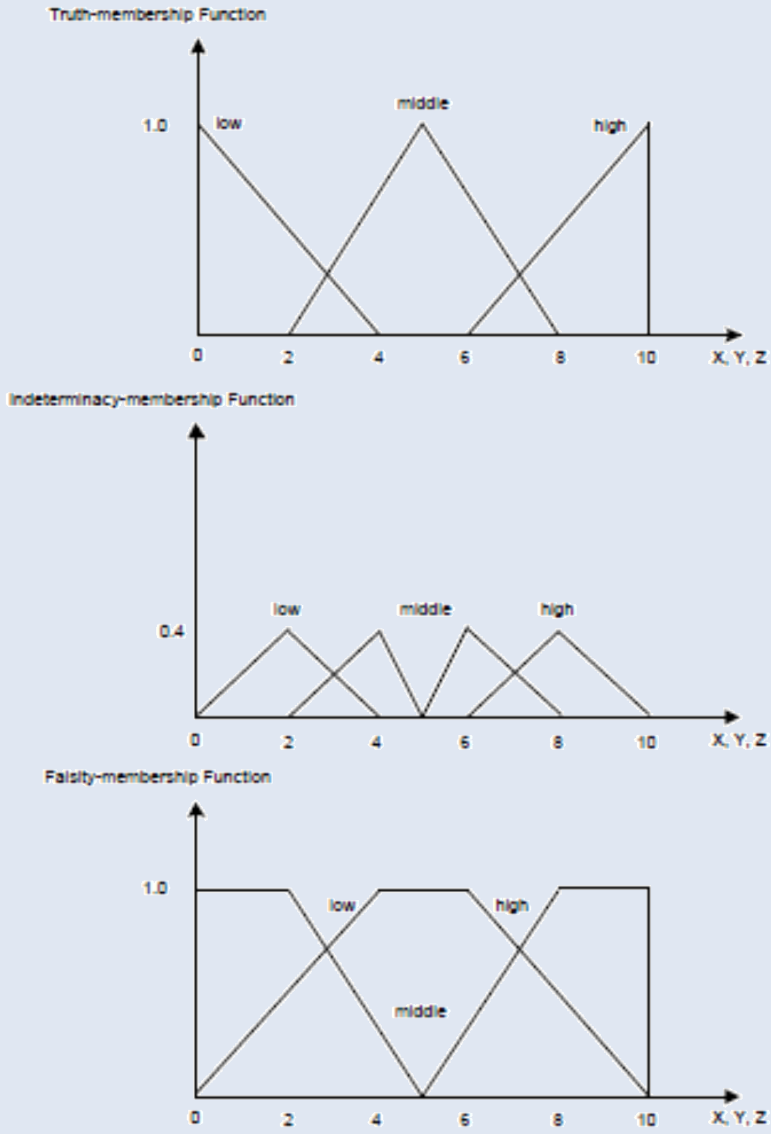


Figure: 12 Membership functions of inputs.

1. Chromosome: The genes of each chromosome are 108 real numbers (there are 108 parameters in the neutrosophic rule base) which are initially generated randomly in the given range. So each chromosome is a vector of 108 real numbers.

2. Fitness function: The fitness function is defined as

$$E = 1/2 \sum_{j=1}^{m} (d_i - o_i)^2 \quad (4.3)$$

3. Elitism: The tournament selection is used in the elitism process.

4. Crossover: The system will randomly select two parents among the population, then randomly select the number of cross points, and simply exchange the corresponding genes among these two parents to generate a new generation.

5. Mutation: For each individual in the population, the system will randomly select genes in the chromosome and replace them with randomly generated real numbers in the given range.

Simulations

There are two phases for applying a fuzzy neural network: training and predicting.

In the training phase, we use 150 data entries as training data set.

Each entry consists of three inputs and one expected output.

We tune the performance of the system by adjusting the size of population, the number of generation and probability of crossover and mutation.

Table below gives the part of prediction results with several parameters for output o.

No. of generation = 10000,
no. of population = 100,
probability of crossover = 0.7,
probability of mutation = 0.3.

The maximum error of prediction result is 1.64.

The total prediction error for 150 entries of testing dataset is 19%.

Input x	Input y	Input z	Desired output	Real output o
1	0	1	0	0.51
1	2	5	1	1.71
1	4	7	2	2.59
3	2	9	3	3.52
3	6	7	4	3.81
3	10	7	5	4.92
5	8	9	6	5.43
7	10	7	7	5.90
7	10	9	8	6.45
9	10	9	9	7.36

Table 3: Prediction Result of Neutrosophic Neural Network.

By our observation, designing reasonable neutrosophic membership functions and choosing reasonable training data set which is based on specific application domain can reduce the prediction error a lot.

Here, the example is just for illustration.

Conclusions

We discussed the design of an extensible soft SWS agent and gave one implementation of Intelligent Inference Engine.

The soft SWS agent supports both keyword based discovery as well as capability based discovery of semantic Web services.

The primary motivation of our work is to solve two challenges facing current Web services advertising and discovery techniques.

One is how to locate the registry hosting required Web service description and another is how to find the required Web service with highest QoS in the located registry.

The soft SWS agent solves both these problems efficiently and effectively.

The soft SWS agent is built upon semantic Web, Web services, and soft computing technologies.

The soft SWS agent could be used in WWW, P2P, or Grid infrastructures.

The soft SWS agent is flexible and extensible.

With the evolution of soft computing, more and more technologies can be integrated into the soft SWS agent.

We used specific ontology based semantic Web services for bioinformatics and neutrosophic neural network with genetic algorithm as our study case.

The training time is short and training results are satisfactory.

The soft SWS agent will return the desired semantic Web services based on the entire QoS of semantic Web services.

In the future, we plan to extend the architecture of the soft SWS agent to compute the entire QoS workflow of semantic Web services to facilitate the composition and monitoring of complex semantic Web services and apply it to semantic Web-based bioinformatics applications.