Fuzzy set theory

Crisp:

- A logic which demands a binary (0/1) type of handling is termed crisp in the domain of fuzzy set theory.

ex: Temperature is 37°C, The running time of a program is 30sec.

- The term fuzzy refers to things which are not clear.

Is water colorless? crisp → yes / → No

Is Ram honest

fuzzy → Extremely honest (1). → very honest (0.8) → honest at times (0.4) → Extremely dishonest (0)

Crisp sets:

Universe of discourse → It is the set which, with reference to a particular context, contains all possible elements having the same characteristics & from which sets can be formed.

→ Universal set is denoted by E.

ex: set of all students in a university.

Set: A set is a well defined collection of objects. Here, well defined means the object either belongs to or does not belongs to the set.

ex: $A = \{Gandhi, Bose, Nehru\}$

$A = \{x \mid P(x)\}$

$P(x)$ stands for the property P to be satisfied by the member x.

ex $A = \{x \mid x \text{ is an odd number}\}$

Venn diagram: Pictorial representation of a set.

E = universal set.

A = Set of female students.

Membership: An element x is said to a member of set A if x blongs to the set A.

$x \in A \quad x \notin B$

ex: $A = \{4,5,6,7,8\}$ $x = 3$ $x \notin A$ $x = 4$. $x \in A$

Cardinality: The no of elements in a set called its cardinality.

ex: $A = \{4,5,6,7\}$ $|A| = 4$

Family of sets: A set whose members are sets themselves.

$A = \{\{1,3,5\}, \{2,4\}\}$

Null set / Empty set: set has no elements.

ex: ϕ or $\{\}$ $|\phi| = 0$

Singleton set: A set with a single element.

ex $A = \{a\}$ then $|A| = 1$

Subset: A is said to be a subset of B if A is fully contained in B, that is, every element of A is in B.

$A \subset B$ A is a proper subset of B

$A \subseteq B$ A is a improper subset of B.

Superset: A is said to be a superset of B if every element of B is contained in A.

ex: $A = \{3,4\}$ $B = \{3,4,5\}$ $A \subset B$

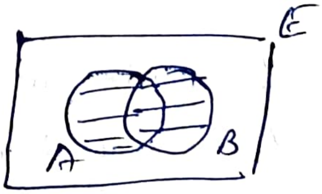
Power set: set of all possible subsets of a given set.

Let $A = \{3,4\}$ $P(A) = \{\phi, \{3\}, \{4\}, \{3,4\}\}$ $|P(A)| = 2^2 = 4$

Operations on crisp sets

Union $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

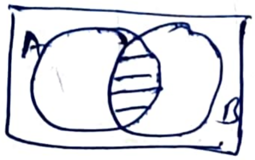
ex: $A = \{a,b,c,1,2\}$ $B = \{1,2,3,a,c\}$ $A \cup B = \{a,b,c,1,2,3\}$



2 **Intersection:** $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

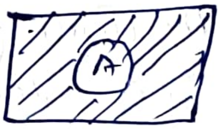
$A \cap B = \phi$ Disjoint sets.

$A = \{a, b, c, 1, 2\}$ $B = \{1, 2, 3, a, c\}$ $A \cap B = \{a, c, 1, 2\}$



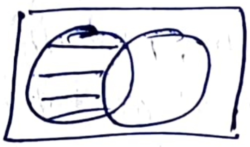
Compliment: $A^c = \{x \mid x \notin A, x \in E\}$

$\cancel{E} = \{1, 2, 3, 4, 5, 6, 7\}$ & $A = \{5, 4, 3\}$ $A^c = \{1, 2, 6, 7\}$



Difference: $A - B = \{x \mid x \in A \ \& \ x \notin B\}$

$A = \{a, b, c, d, e\}$ $B = \{b, d\}$ $A - B = \{a, c, e\}$



Properties of CRISP sets:

Commutativity $A \cup B = B \cup A$
$A \cap B = B \cap A$

Associativity $(A \cup B) \cup C = A \cup (B \cup C)$
$(A \cap B) \cap C = A \cap (B \cap C)$

Distributivity $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Law of absorption $A \cup (A \cap B) = A$
$A \cap (A \cup B) = A$

Law of contradiction
$A \cap A^c = \phi$

Idempotence: $A \cup A = A$
$A \cap A = A$

Identity $A \cup \phi = A$ $A \cup E = E$
$A \cap E = A$ $A \cap \phi = \phi$

Transitivity $A \subseteq B, B \subseteq C \Rightarrow A \subseteq C$

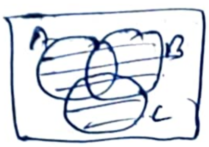
Involution $(A^c)^c = A$

Law of excluded middle $A \cup A^c = E$

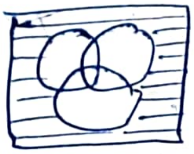
Demorgan's law
$(A \cup B)^c = A^c \cap B^c$
$(A \cap B)^c = A^c \cup B^c$

Ex: Given three sets A, B, & C Prove De Morgan's Law using Venn diagram.

(I) $(A\cup B\cup C)^c = A^c\cap B^c\cap C^c$ (II) $(A\cap B\cap C)^c = A^c\cup B^c\cup C^c$



$A\cup B\cup C$


$(A\cup B\cup C)^c$

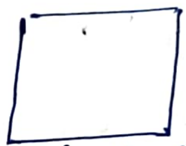

A^c

B^c


C^c


$A^c\cap B^c\cap C^c$


$(A\cap B\cap C)^c$


$A^c\cup B^c\cup C^c$

ex 2: E = all students enrolled in the university cricket club.
A = male students, B = Bowlers, C = batsmen.

Draw venn diagram for
1) female students
2) only bowler Not a batsmen
3) Fem. can both bow & bat.

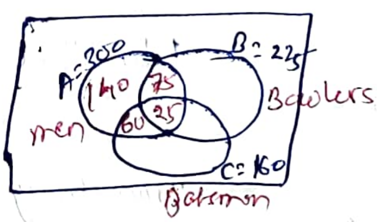
female students


Bowlers who are not batsmen


F students who can both bowl & bat.

ex 3 $|E| = 600$ $|A| = 300$ $|B| = 225$; $|C| = 160$. Also, the no of male students who are bowlers $(A\cap B)$ be 100, 25 of whom are batsmen too $(A\cap B\cap C)$, & the total no of male students who are batsmen $(A\cap C)$ be 85.

Determine (1) female students 2) Not bowlers 3) Not batsmen
(4) Female students who can bowl.



1) No of female students $= |A^c| = |E| - |A| = 600 - 300 = 300$

2) No of students who are not bowlers $= |B^c| = |E| - |B| = 600 - 225 = 375$

3) N of students who are not batsmen $= |C^c| = |E| - |C| = 600 - 160 = 440$

4) N of female students who can bowl

$|A^c\cap B| = 125$

Ex: $A = \{a, b, c, d, e\}$ $A_1 = \{a, b\}$ $A_2 = \{c, d\}$ $A_3 = \{e\}$ (5)

$|A| = 5$ & $\sum_{i=1}^{3} |A_i| = 2+2+1 = 5$

Rule of inclusion & Exclusion:

Given A to be a covering of n sets $A_1, A_2, \ldots A_n$

for $n = 2$ $|A| = |A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

, $n = 3$ $|A| = |A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_2 \cap A_3|$
$- |A_1 \cap A_3| + |A_1 \cap A_2 \cap A_3|$

Ex: Given $|E| = 100$ E is a set of students who have choosen subjects from different stream in CS, it is found that 32 study subjects choosen from the CN stream, 20 from Multimedia (MM) & 45 from software systems (SS). Also, 15 study subjects from both CN & SS. 7 from both MM & SS, and 30 do not study any subjects choosen from either of three subjects.

Find the no of students who study subjects belonging to all three subjects.

$A = CC$, $MM = B$ $C = SS$ $|A \cap B \cap C| = ?$

$|A^c \cap B^c \cap C^c| = 30$, $|(A \cup B \cup C)^c| = 30$ using demorgan's law

$|E| - |A \cup B \cup C| = 30$

$|A \cup B \cup C| = E - 30 = 70$

$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$

$70 = 32 + 20 + 45 - 15 - 7 - 10 + |A \cap B \cap C|$

$70 - 65 = |A \cap B \cap C|$

$5 = |A \cap B \cap C|$

Partition & Covering

① Partition:

A partition on A is defined to be a set of non-empty subsets A_i, each of which is pairwise disjoint & whose union yields the original set A.

Partition on A indicated as $\pi(A)$, is therefore

(i) $A_i \cap A_j = \phi$ for each pair $(i,j) \in I$, $i \neq j$

(ii) $\bigcup_{i \in I} A_i = A$

→ The members of A_i of partition are known as blocks.

Partition on set A

ex: $A = \{a, b, c, d, e\}$. $A_1 = \{a, b\}$ $A_2 = \{c, d\}$ $A_3 = \{e\}$

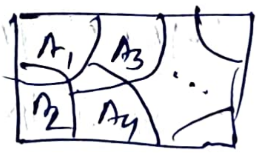
$A_1 \cap A_2 = \phi$ $A_1 \cap A_3 = \phi$; $A_2 \cap A_3 = \phi$

$A_1 \cup A_2 \cup A_3 = A = \{a, b, c, d, e\}$

Hence $\{A_1, A_2, A_3\}$ is a partition on A.

Covering

A covering on set A is defined to be a set of nonempty subsets A_i, whose union yields the original set A. The nonempty subsets needs not be disjoint.



ex: $A = \{a, b, c, d, e\}$ $A_1 = \{a, b\}$ $A_2 = \{b, c, d\}$ $A_3 = \{d, e\}$

$A_1 \cap A_2 = \{b\}$ $A_1 \cap A_3 = \phi$ $A_2 \cap A_3 = \{d\}$

$A_1 \cup A_2 \cup A_3 = \{a, b, c, d, e\} = A$ is a covering.

Rule of Addition:

Given a partition on A where A_i, $i = 1, 2, \ldots, n$ are non-empty subsets then,

$$|A| = \left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i|$$

Fuzzy sets

- supports a flexible sense of membership of elements to a set.
- many degree of membership (blw 0 & 1) are allowed.
- A membership func^n $\mu_A^{(x)}$ is associated with a fuzzy set $\tilde{A}$ such that the function maps every element of the universe of discourse X (or the reference set) to the interval $[0, 1]$

formally, the mapping is written as $\mu_A^x : X \rightarrow [0, 1]$.

If X is a universe of discourse & x is a particular element of X, then a fuzzy set A defined on X may be written as a collection of ordered pairs.

$$A = \{ (x, \mu_A^{(x)}(x)), x \in X \}$$

where each pair $(x, \mu_A^{(x)})$ is called a singleton.

→ An alternate def^n which indicates a fuzzy set as a union of all $\mu_A(x)/x$ singletons is given by:

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i \quad \text{.. in the discrete case}$$

$$A = \int_x \mu_A(x)/x \quad \text{in the continuous case.}$$

ex:

$X = \{ g_1, g_2, g_3, g_4, g_5 \}$ be the reference set of students.

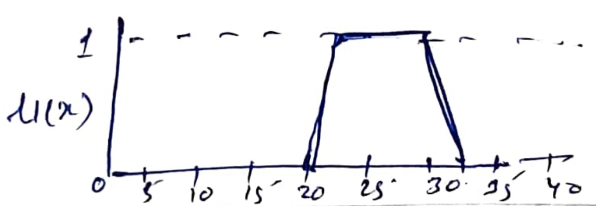
Let $\tilde{A}$ be the fuzzy set of smart students, where "smart" is a fuzzy linguistic term.

honest

$$\tilde{A} = \{ (g_1, 0.4), (g_2, 0.5), (g_3, 1), (g_4, 0.9), (g_5, 0.8) \}$$

Membership function :-

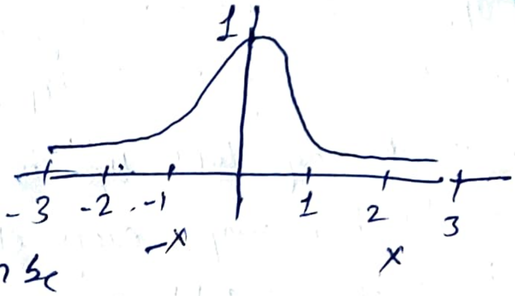
- membership function values need not be described by discrete values. Quite often, these turn out to be as described by a continuous func^n.



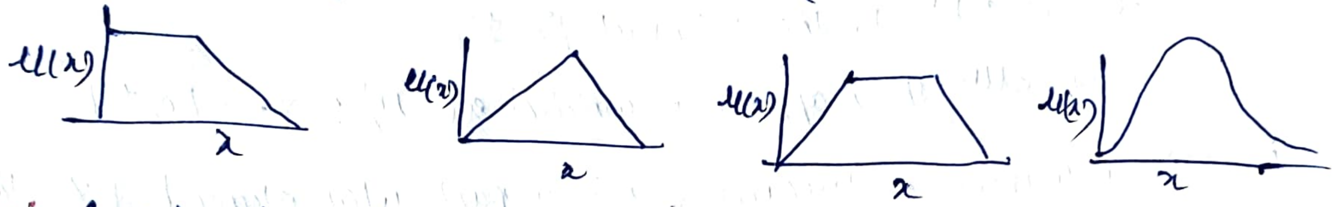
continuos membership funcn for "cool"

→ A membership function can also be given mathematically as

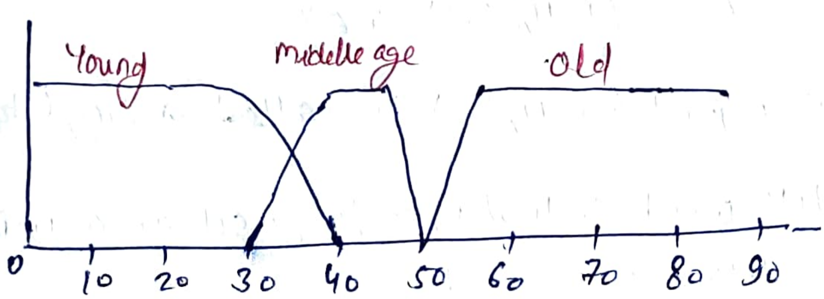
$$\mu_A(x) = \frac{1}{(1+x)^2}$$



→ Different shapes of membership functions can be



Ex: Consider the set of people in the age groups-
0-10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70 & above.



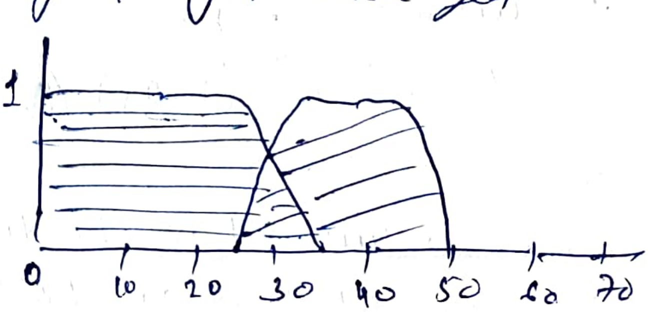
Basic fuzzy set operations:

X = universe of discourse $\tilde{A}$ & $\tilde{B}$ are fuzzy sets with $\mu_A(x)$ & $\mu_B(x)$

Union:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

ex: $\tilde{A}$ = Set of young people. $\tilde{B}$ = set of middle aged people
$\tilde{A} \cup \tilde{B}$, be fuzzy set of "young or middle aged"



In its discrete form, for x_1, x_2, x_3

$\tilde{A} = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0)\}$ & $\tilde{B} = \{(x_1, 0.8), (x_2, 0.2), (x_3, 1)\}$

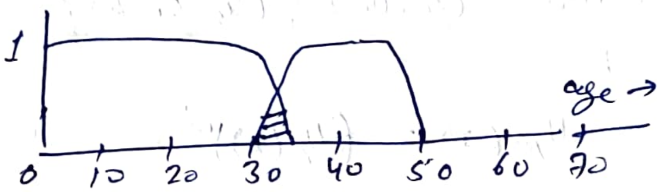
$\tilde{A} \cup \tilde{B} = \{(x_1, 0.8), (x_2, 0.7), (x_3, 1)\}$

Intersection:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

for $\tilde{A}$ = young $\tilde{B}$ = middle aged $\tilde{A} \cap \tilde{B}$ = young & middle aged

$\tilde{A} \cap \tilde{B}$



A

In the discrete form for x_1, x_2, x_3

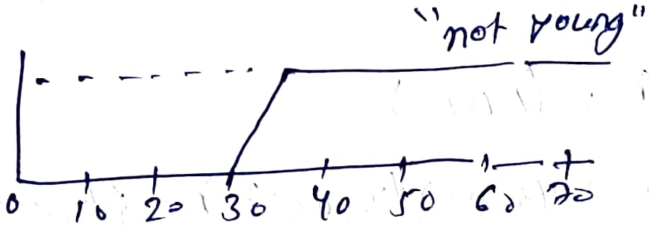
$\tilde{A} = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0)\}$ & $\tilde{B} = \{(x_1, 0.8), (x_2, 0.2), (x_3, 1)\}$

$\tilde{A} \cap \tilde{B} = \{(x_1, 0.5), (x_2, 0.2), (x_3, 0)\}$

Complement:

$$\mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x)$$

$\tilde{A}^c$



$\tilde{A} = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0)\}$

$\tilde{A}^c = \{(x_1, 0.5), (x_2, 0.3), (x_3, 1)\}$

Product of two fuzzy sets:

$$\mu_{\tilde{A}.\tilde{B}}(x) = \mu_{\tilde{A}}(x) \cdot \mu_{\tilde{B}}(x)$$

$\tilde{A} = \{(x_1, 0.2), (x_2, 0.8), (x_3, 0.4)\}$ $\tilde{B} = \{(x_1, 0.4), (x_2, 0), (x_3, 0.1)\}$

$\tilde{A}.\tilde{B} = \{(x_1, 0.08), (x_2, 0), (x_3, 0.04)\}$

Equality:

$$\tilde{A} = \tilde{B} \text{ iff } \mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x)$$

ex: $A = \{(x_1, 0.2), (x_2, 0.8)\}$ $B = \{(x_1, 0.6), (x_2, 0.8)\}$ $\tilde{C} = \{(x_1, 0.2), (x_2, ...$

$$\tilde{A} \neq \tilde{B} \quad \tilde{A} = \tilde{C}$$

Properties
Commutati...

Product of a fuzzy set with a crisp number

$$\mu_{a.\tilde{A}}(x) = a.\mu_{\tilde{A}}(x)$$

ex: $\tilde{A} = \{(x_1, 0.4), (x_2, 0.6), (x_3, 0.8)\}$ $a = 0.3$

$a.\tilde{A} = \{(x_1, 0.12), (x_2, 0.18), (x_3, 0.24)\}$

Power of a fuzzy set:

$$\mu_{\tilde{A}^\alpha}(x) = \{\mu_{\tilde{A}}(x)\}^\alpha$$

→ Raising a fuzzy set to its second power is called concentration (CON) & taking the squre root is called Dilation (DIL).

ex: $\tilde{A} = \{(x_1, 0.4), (x_2, 0.2), (x_3, 0.7)\}$, $\alpha = 2$

$\mu_{\tilde{A}^2}(x) = \{\mu_{\tilde{A}}(x)\}^2$

$(\tilde{A})^2 = \{(x_1, 0.16), (x_2, 0.04), (x_3, 0.49)\}$

Difference:

$$\tilde{A} - \tilde{B} = (\tilde{A} \cap \tilde{B}^c)$$

ex: $\tilde{A} = \{(x_1, 0.2), (x_2, 0.5), (x_3, 0.6)\}$ $\tilde{B} = \{(x_1, 0.1), (x_2, 0.4), (x_3, 0.5)\}$

$\tilde{B}^c = \{(x_1, 0.9), (x_2, 0.6), (x_3, 0.5)\}$

$\tilde{A} - \tilde{B} = \tilde{A} \wedge \tilde{B}^c = \{(x_1, 0.2), (x_2, 0.5), (x_3, 0.5)\}$

Disjunctive Sum:

$$\tilde{A} \oplus \tilde{B} = (\tilde{A}^c \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{B}^c)$$

$\tilde{A} = \{(x_1, 0.4), (x_2, 0.8), (x_3, 0.6)\}$ $\tilde{B} = \{(x_1, 0.2), (x_2, 0.6), (x_3, 0.9)\}$

$\tilde{A}^c = \{(x_1, 0.6), (x_2, 0.2) (x_3, 0.4)\}$ $\tilde{B}^c = \{(x_1, 0.8), (x_2, 0.4), (x_3, 0.1)\}$

$\tilde{A}^c \cap \tilde{B} = \{(x_1, 0.2) (x_2, 0.2), (x_3, 0.4)\}$ $\tilde{A} \cap \tilde{B}^c = \{(x_1, 0.4), (x_2, 0.4), (x_3, 0.1)\}$

$\tilde{A} \oplus \tilde{B} = \{(x_1, 0.4), (x_2, 0.4) (x_3, 0.4)\}$

Properties of Fuzzy sets:

3. Commutativity

$\tilde{A} \cup \tilde{B} = \tilde{B} \cup \tilde{A}$

$\tilde{A} \cap \tilde{B} = \tilde{B} \cap \tilde{A}$

Associativity:

$\tilde{A} \cup (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cup \tilde{C}$

$\tilde{A} \cap (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cap \tilde{C}$

Distributivity:

$\tilde{A} \cup (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cap (\tilde{A} \cup \tilde{C})$

$\tilde{A} \cap (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{C})$

Idempotence $\tilde{A} \cup \tilde{A} = \tilde{A}$, $\tilde{A} \cap \tilde{A} = \tilde{A}$

Identity $\tilde{A} \cup \phi = \tilde{A}$, $\tilde{A} \cup X = X$

$\tilde{A} \cap \phi = \phi$ $\tilde{A} \cap X = \tilde{A}$

Transitivity: If $\tilde{A} \subseteq \tilde{B}$, $\tilde{B} \subseteq \tilde{C}$ then $\tilde{A} \subseteq \tilde{C}$

Involution: $(\tilde{A}^c)^c = \tilde{A}$

De morgen's Law

$(\tilde{A} \cup \tilde{B})^c = (\tilde{A}^c \cap \tilde{B}^c)$

$(\tilde{A} \cap \tilde{B})^c = (\tilde{A}^c \cup \tilde{B}^c)$

ex: The task is to recognize English alphabetical characters (F, E, X, Y, I, T) in an image processing system, Define two fuzzy sets $\tilde{I}$ & $\tilde{F}$ to represents the identification of characters I & F

$\tilde{I} = \{(F,0.4), (E,0.3), (X,0.1), (Y,0.1), (I,0.9), (T,0.8)\}$

$\tilde{F} = \{(F,0.99), (E,0.8), (X,0.1), (Y,0.2), (I,0.5), (T,0.5)\}$

Find the following:

a) I) $\tilde{I} \cup \tilde{F}$ II) $\tilde{I} - \tilde{F}$ III) $\tilde{F} \cup \tilde{F}^c$

b) verify De morgen's Law $(\tilde{I} \cup \tilde{F})^c = \tilde{I}^c \cap \tilde{F}^c$

(a) I) $\tilde{I} \cup \tilde{F} = \{(F,0.99), (E,0.8), (X,0.1), (Y,0.2), (I,0.9), (T,0.8)\}$

II) $\tilde{I} - \tilde{F} = I \cap F^c$

$= \{(F,0.01), (E,0.2), (X,0.1), (Y,0.1), (I,0.5), (T,0.5)\}$

III) $\tilde{F} \cup \tilde{F}^c = \{(F,0.99), (E,0.8), (X,0.9), (Y,0.8), (I,0.5), (T,0.5)\}$

(b) $(\tilde{I} \cup \tilde{F})^c = \tilde{I}^c \cap \tilde{F}^c$

$(\tilde{I} \cup \tilde{F})^c = \{(F,0.01), (E,0.2), (X,0.9), (Y,0.8), (I,0.1), (T,0.2)\}$

$I^c = \{(F,0.6), (E,0.7), (X,0.9), (Y,0.9), (I,0.1), (T,0.2)\}$

$F^c = \{(F,0.01), (E,0.2), (X,0.9), (Y,0.8), (I,0.5), (T,0.5)\}$

$\tilde{I}^c \cap \tilde{F}^c = \{(F,0.01), (E,0.2), (X,0.9), (Y,0.8), (I,0.1), (T,0.2)\}$

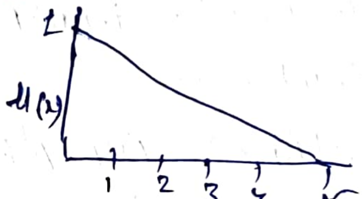
Qu) Consider the fuzzy sets $\tilde{A}$ & $\tilde{B}$ defined on the interval $X = [0, \infty)$ of real numbers, by the membership grade function

$$\mu_{\tilde{A}}(x) = \frac{x}{x+1}, \quad \mu_{\tilde{B}}(x) = 2^{-x}$$

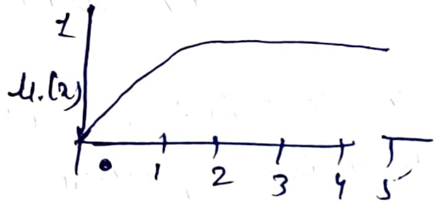
Determine the mathematical formulae & graphs of membership grade function of each of the following sets.

a) A', B^c b) $A \cup B$ c) $A \cap B$ d) $(A \cup B)^c$

a) $\mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x) = 1 - \frac{x}{x+1} = \frac{1}{x+1}$

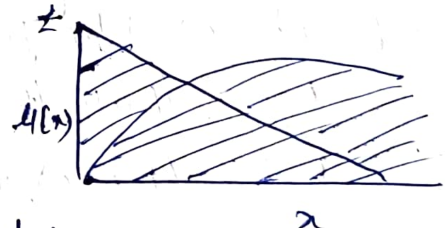


$\mu_{\tilde{B}^c}(x) = 1 - \mu_{\tilde{B}}(x) = 1 - 2^{-x} = \frac{2^x - 1}{2^x}$



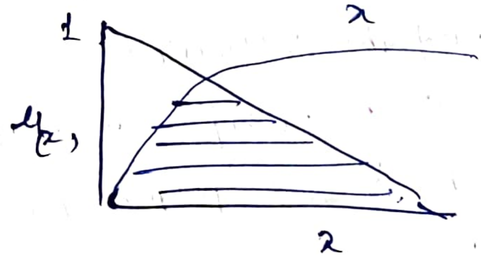
b) $\mu_{\tilde{A} \cup \tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$

$= \max\left(\frac{x}{x+1}, 2^{-x}\right)$



c) $\mu_{\tilde{A} \cap \tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$

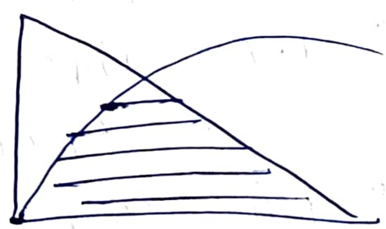
$= \min\left(\frac{x}{x+1}, 2^{-x}\right)$



d) $\mu_{(\tilde{A} \cup \tilde{B})^c}(x) = \mu_{\tilde{A}^c \cap \tilde{B}^c}(x)$

$= \min(\mu_{\tilde{A}^c}(x), \mu_{\tilde{B}^c}(x))$

$= \min\left(\frac{1}{x+1}, \frac{2^x - 1}{2^x}\right)$



CRISP Relations

Cartesian Products

The Cartesian Product of two sets A & B denoted by $A \times B$ is the set of all ordered pairs such that the first element in the pair belongs to A & the second element belongs to B

i.e.

$$A \times B = \{(a,b) \mid a \in A,\ b \in B\}$$

→ If $A \neq B$ & A & B are non empty sets then $A \times B \neq B \times A$

The cartesian Product could be extended to n number of sets

$$\overset{n}{\underset{i=1}{\times}} A_i = \{(a_1, a_2, a_3, \ldots, a_n) \mid a_i \in A_i \text{ for every } i = 1,2,\ldots n\}$$

observe that

$$\left| \overset{n}{\underset{i=1}{\times}} A_i \right| = \prod_{i=1}^{n} |A_i|$$

ex:

Given $A_1 = \{a,b\}$ $A_2 = \{1,2\}$; $A_3 = \{\alpha\}$

$A_1 \times A_2 = \{(a,1),(b,1),(a,2),(b,2)\}$ $|A_1 \times A_2| = 4$ $|A_1| = |A_2| = 2$

$|A_1 \times A_2| = |A_1| \cdot |A_2|$

also $A_1 \times A_2 \times A_3 = \{(a,1,\alpha),(a,2,\alpha),(b,1,\alpha),(b,2,\alpha)\}$

$|A_1 \times A_2 \times A_3| = 4 = |A_1| \cdot |A_2| \cdot |A_3|$

ex 2 Given $X = \{1,2,3,4\}$

$X \times X = \{(1,1),(1,2),(1,3),(1,4),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(3,4),$
$(4,1),(4,2),(4,3),(4,4)\}$

Let the R be defined as $R = \{(x,y) \mid y = x+1,\ x,y \in X\}$

$R = \{(1,2),(2,3),(3,4)\}$

The Relation matrix R is given by

$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Operations on Relation :-

Union $R \cup S$ $\quad$ $R\cup S(x,y) = \max\{R(x,y), S(x,y)\}$

Intersection $\quad$ $R \cap S(x,y) = \min(R(x,y), S(x,y))$

Complement $\quad$ $\bar{R}(x,y) = 1 - R(x,y)$

Composition of relations

→ Given R to be a relation on X, Y & S to be a relation on Y, Z then RoS is composition of relation on X, Z defined as:

$$R \circ S = \{(x,z) \mid (x,z) \in X \times Z,\ \exists\, y \in Y \text{ such that } (x,y) \in R \text{ \& } (y,z) \in S\}$$

→ A common form of the composition relation is the max-min composition.

Max-min Composition :-

~~Let R, S be defined on the sets {1,3,5} × {1,3,5}~~

Given the relation matrices of the relation R & S, the max-min composition is defined as

$$T = R \circ S$$

$$T(x,z) = \max_{y \in Y}\left(\min(R(x,y), S(y,z))\right)$$

ex: Let R, S be defined on the sets $\{1,3,5\} \times \{1,3,5\}$

$$R: \{(x,y) \mid y = x+2\},\quad S: \{(x,y) \mid x<y\}$$

$$R = \{(1,3), (3,5)\},\quad S = \{(1,3), (1,5), (3,5)\}$$

The relation matrices are

$$R: \begin{array}{c|c} & \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad S: \begin{array}{c|c} & \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

using max-min composition

$$R\circ S \quad \begin{array}{c|c} & \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$RoS(1,1) = \max\{\min(0,0), \min(1,0), \min(0,0)\} = \max(0,0,0) = 0$

$RoS(1,3) = \max\{0,0,0\} = 0$

$RoS(1,5) = \max\{0,1,0\} = 1$

$RoS(3,1) = 0$, $RoS(3,3) = RoS(3,5) = RoS(5,1) = RoS(5,3) = RoS(5,5) = 0$

RoS from the relation matrix Fs $\{(1,5)\}$

$$SoR = \begin{array}{c} 1 \\ 3 \\ 5 \end{array}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Fuzzy Relation:

- Fuzzy relation is a fuzzy set defined on the cartesian product of crisp sets $X_1, X_2, \ldots X_n$ where the n-tuples $(x_1, x_2, \ldots, x_n)$ may have varying degrees of membership within the relation. The membership values indicate the strength of the relation b/w the tuples.

Ex: Let R be the fuzzy relation b/w sets X_1 & X_2 where X_1 is the set of diseases & X_2 is the set of symptoms.

$X_1 = \{$ typhoid, viral fever, common cold $\}$

$X_2 = \{$ running nose, high temperature, shivering $\}$

The fuzzy relation R may be defined as.

	Running Nose	High temperature	Shivering
Typhoid	0.1	0.9	0.8
viral fever	0.2	0.9	0.7
Common cold	0.9	0.4	0.6

Fuzzy Cartesian Product:

Let $\tilde{A}$ be a fuzzy set defined on the universe X & $\tilde{B}$ be a fuzzy set defined on the universe Y, the cartesian product b/w the fuzzy sets $\tilde{A}$ & $\tilde{B}$ indicated as $\tilde{A} \times \tilde{B}$ & resulting in a fuzzy relation $\tilde{R}$ given by

$$\tilde{R} = \tilde{A} \times \tilde{B} \subset X \times Y$$

where $\tilde{R}$ has its membership function given by

$$\mu_{\tilde{R}}(x, y) = \mu_{\tilde{A} \times \tilde{B}}(x, y)$$
$$= \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y))$$

ex:

$\tilde{A} = \{(x_1, 0.2), (x_2, 0.7), (x_3, 0.4)\}$ $\tilde{B} = \{(y_1, 0.5), (y_2, 0.6)\}$ be two fuzzy sets defined on the universes of discourse $X = \{x_1, x_2, x_3\}$ & $Y = \{y_1, y_2\}$ respectively. Then the fuzzy relation $\tilde{R}$ resulting out of the fuzzy cartesian product $\tilde{A} \times \tilde{B}$ is given by

$$\tilde{R} = \tilde{A} \times \tilde{B} = \begin{array}{c c} & \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.2 & 0.2 \\ 0.5 & 0.6 \\ 0.4 & 0.4 \end{bmatrix} \end{array}$$

Since

$\tilde{R}(x_1, y_1) = \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{B}}(y_1)) = \min(0.2, 0.5) = 0.2$

$R(x_1, y_2) = \min(0.2, 0.6) = 0.2$ $\tilde{R}(x_3, y_1) = \min(0.4, 0.5) = 0.4$

$R(x_2, y_1) = \min(0.7, 0.5) = 0.5$ $\tilde{R}(x_3, y_2) = \min(0.4, 0.6) = 0.4$

$R(x_2, y_2) = \min(0.7, 0.6) = 0.6$

Operations on Fuzzy Relations

Let $\tilde{R}$ & $\tilde{S}$ be fuzzy relations on $X \times Y$

Union $\mu_{\tilde{R} \cup \tilde{S}}(x,y) = \max(\mu_{\tilde{R}}(x,y); \mu_{\tilde{S}}(x,y))$

Intersectⁿ $\mu_{\tilde{R} \cap \tilde{S}}(x,y) = \min(\mu_{\tilde{R}}(x,y), \mu_{\tilde{S}}(x,y))$

Complement $\mu_{\tilde{R}^c}(x,y) = 1 - \mu_{\tilde{R}}(x,y)$

Composition of relation

$$\mu_{\tilde{R} \circ \tilde{S}}(x,z) = \max(\min(\mu_{\tilde{R}}(x,y), \mu_{\tilde{S}}(y,z)))$$

ex: $X = \{x_1, x_2, x_3\}$ $Y = \{y_1, y_2\}$ $Z = \{z_1, z_2, z_3\}$

Let $\tilde{R}$ be a fuzzy relation. Let $\tilde{S}$ be a fuzzy relation

$$\begin{array}{c|cc} & y_1 & y_2 \\ \hline x_1 & 0.5 & 0.1 \\ x_2 & 0.2 & 0.9 \\ x_3 & 0.8 & 0.6 \end{array} \qquad \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ \hline y_1 & 0.6 & 0.4 & 0.7 \\ y_2 & 0.5 & 0.8 & 0.9 \end{array}$$

Then $R \circ S$ by max-min composition yields

$$R \circ S = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ \hline x_1 & 0.5 & 0.4 & 0.5 \\ x_2 & 0.5 & 0.8 & 0.9 \\ x_3 & 0.6 & 0.6 & 0.7 \end{array}$$

$\mu_{\tilde{R} \circ \tilde{S}}(x_1, z_1) = \max(\min(0.5, 0.6), \min(0.1, 0.5)) = \max(0.5, 0.1) = 0.5$

$\mu_{\tilde{R} \circ \tilde{S}}(x_1, z_2) = \max(\min(0.5, 0.4), \min(0.1, 0.8)) = \max(0.4, 0.1) = 0.4$

$\mu_{R \circ S}(x_1, z_3) = \max(0.5, 0.1) = 0.5$

$(x_2, z_1) = \max(0.2, 0.5) = 0.5$

$(x_2, z_2) = \max(0.2, 0.8) = 0.8$

$(x_2, z_3) = \max(0.2, 0.9) = 0.9$

$(x_3, z_1) = \max(0.6, 0.5) = 0.6$

$(x_3, z_2) = \max(0.4, 0.6) = 0.6$

$(x_3, z_3) = \max(0.7, 0.6) = 0.7$

18

ex 3 $P = \{P_1, P_2, P_3, P_4\}$ of four varieties of paddy plants, set $D = \{D_1, D_2, D_3, D_4\}$ of the various diseases affecting the plants & $S = \{S_1, S_2, S_3, S_4\}$ be the common symptoms of the diseases.

Let $\tilde{R}$ be a relation on $P \times D$ & $\tilde{S}$ be a relation on $D \times S$

$$\tilde{R} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & D_3 & D_4 \end{array} \\ \begin{bmatrix} 0.6 & 0.6 & 0.9 & 0.8 \\ 0.1 & 0.2 & 0.9 & 0.8 \\ 0.9 & 0.3 & 0.4 & 0.8 \\ 0.9 & 0.8 & 0.1 & 0.2 \end{bmatrix} \end{array} \qquad \tilde{S} = \begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{bmatrix} 0.1 & 0.2 & 0.7 & 0.9 \\ 1 & 1 & 0.4 & 0.6 \\ 0 & 0 & 0.5 & 0.9 \\ 0.9 & 1 & 0.8 & 0.2 \end{bmatrix} \end{array}$$

Obtain the association of the plants with the different symptoms of the diseases using max-min composition.

Soln

$$R \circ S = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{bmatrix} 0.8 & 0.8 & 0.8 & 0.9 \\ 0.8 & 0.8 & 0.8 & 0.9 \\ 0.8 & 0.8 & 0.8 & 0.9 \\ 0.8 & 0.8 & 0.7 & 0.9 \end{bmatrix} \end{array}$$

Chapter 7

Fuzzy Systems

Logic is the science of reasoning. Symbolic or *mathematical* logic has turned out to be a powerful computational paradigm. Not only does symbolic logic help in the description of events in the real world but has also turned out to be an effective tool for inferring or deducing information from a given set of facts.

Just as mathematical sets have been classified into crisp sets and fuzzy sets (Refer Chapter 6), logic can also be broadly viewed as *crisp logic* and *fuzzy logic.* Just as crisp sets survive on a 2-state membership (0/1) and fuzzy sets on a multistate membership [0–1], crisp logic is built on a 2-state truth value (True/False) and fuzzy logic on a multistate truth value (True/False/very True/partly False and so on.)

We now briefly discuss crisp logic as a prelude to fuzzy logic.

7.1 CRISP LOGIC -Defn

Consider the statements "Water boils at 90°C" and "Sky is blue". An agreement or disagreement with these statements is indicated by a "True" or "False" value accorded to the statements. While the first statement takes on a value *false*, the second takes on a value *true*.

Thus, a statement which is either 'True' or 'False' but not both is called a *proposition*. A proposition is indicated by upper case letters such as P, Q, R and so on.

Example: P: Water boils at 90°C.
Q: Sky is blue.

are propositions.

A simple proposition is also known as an *atom*. Propositions alone are insufficient to represent phenomena in the real world. In order to represent complex information, one has to build a sequence of propositions linked using *connectives* or *operators*. Propositional logic recognizes five major operators as shown in Table 7.1.

Table 7.1 Propositional logic connectives

Symbol	Connective	Usage	Description
$\wedge$	and	$P \wedge Q$	P and Q are true.
$\vee$	or	$P \vee Q$	Either P or Q is true.
$\neg$ or $\sim$	not	$\sim P$ or $\neg P$	P is not true.
$\Rightarrow$	implication	$P \Rightarrow Q$	P implies Q is true.
$=$	equality	$P = Q$	P and Q are equal (in truth values) is true.

Observe that $\wedge$, $\vee$, $\Rightarrow$, and $=$ are 'binary' operators requiring two propositions while $\sim$ is a 'unary' operator requiring a single proposition. $\wedge$ and $\vee$ operations are referred to as *conjunction* and *disjunction* respectively. In the case of $\Rightarrow$ operator, the proposition occurring before the '$\Rightarrow$' symbol is called as the *antecedent* and the one occurring after is called as the *consequent*.

The semantics or meaning of the logical connectives are explained using a *truth table*. A truth table comprises rows known as *interpretations*, each of which evaluates the logical formula for the given set of truth values. Table 7.2 illustrates the truth table for the five connectives.

Table 7.2 Truth table for the connectives $\wedge$, $\vee$, $\sim$, $\Rightarrow$, $=$

P	Q	$P \wedge Q$	$P \vee Q$	$\sim P$	$P \Rightarrow Q$	$P = Q$
T	T	T	T	F	T	T
T	F	F	T	F	F	F
F	F	F	F	T	T	T
F	T	F	T	T	T	F

T : True, F : False

A logical formula comprising n propositions will have 2^n interpretations in its truth table. A formula which has all its interpretations recording true is known as a *tautology* and the one which records false for all its interpretations is known as *contradiction*.

Example 7.1

Obtain a truth table for the formula $(P \vee Q) \Rightarrow (\sim P)$. Is it a tautology?

Solution

The truth table for the given formula is

P	Q	$P \vee Q$	$\sim P$	$P \vee Q \Rightarrow \sim P$
T	F	T	F	F
F	T	T	T	T
T	T	T	F	F
F	F	F	T	T

No, it is not a tautology since all interpretations do not record 'True' in its last column.

Example 7.2

Is $((P \Rightarrow Q) \wedge (Q \Rightarrow P)) = (P = Q)$ a tautology?

Solution

P	Q	$P \Rightarrow Q$	$Q \Rightarrow P$	A: $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$	B: $P = Q$	$A = B$
T	F	F	T	F	F	T
F	T	T	F	F	F	T
T	T	T	T	T	T	T
F	F	T	T	T	T	T

Yes, the given formula is a tautology.

Example 7.3

Show that $(P \Rightarrow Q) = (\sim P \vee Q)$

Solution

The truth table for the given formula is

P	Q	A: $P \Rightarrow Q$	$\sim P$	B: $\sim P \vee Q$	$A = B$
T	T	T	F	T	T
T	F	F	F	F	T
F	F	T	T	T	T
T	T	T	T	T	T

Since the last column yields 'True' for all interpretations, it is a tautology.

The logical formula presented in Example 7.3 is of practical importance since $(P \Rightarrow Q)$ is shown to be equivalent to $(\sim P \vee Q)$, a formula devoid of '$\Rightarrow$' connective. This equivalence can therefore be utilised to eliminate '$\Rightarrow$' in logical formulae.

It is useful to view the '$\Rightarrow$' operator from a set oriented perspective. If X is the universe of discourse and A, B are sets defined in X, then propositions P and Q could be defined based on an element $x \in X$ belonging to A or B. That is,

$$P: x \in A$$

$$Q: x \in B \tag{7.1}$$

Here, P, Q are true if $x \in A$ and $x \in B$ respectively, and $\sim P$, $\sim Q$ are true if $x \notin A$ and $x \notin B$ respectively. In such a background, $P \Rightarrow Q$ which is equivalent to $(\sim P \vee Q)$ could be interpreted as

$$(P \Rightarrow Q) : x \notin A \quad \text{or} \quad x \in B \tag{7.2}$$

However, if the '$\Rightarrow$' connective deals with two different universes of discourse, that is, $A \subset X$ and $B \subset Y$ where X and Y are two universes of discourse then the '$\Rightarrow$' connective is represented by the relation R such that

$$R = (A \times B) \cup (\bar{A} \times Y) \tag{7.3}$$

In such a case, $P \Rightarrow Q$ is linguistically referred to as IF A THEN B. The compound proposition $(P \Rightarrow Q) \vee (\sim P \Rightarrow S)$ linguistically referred to as IF A THEN B ELSE C is equivalent to

$$\text{IF } A \text{ THEN } B\ (P \Rightarrow Q)$$

$$\text{IF } \sim A \text{ THEN } C\ (\sim P \Rightarrow S) \tag{7.4}$$

where P, Q, and S are defined by sets A, B, C, $A \subset X$, and B, $C \subset Y$.

7.1.1 Laws of Propositional Logic

Crisp sets as discussed in Section 6.2.2. exhibit properties which help in their simplification.

Similarly, propositional logic also supports the following laws which can be effectively used for their simplification. Given P, Q, R to be the propositions,

(i) *Commutativity*

$$(P \vee Q) = (Q \vee P)$$
$$(P \wedge Q) = (Q \wedge P) \tag{7.5}$$

(ii) *Associativity*

$$(P \vee Q) \vee R = P \vee (Q \vee R)$$
$$(P \wedge Q) \wedge R = P \wedge (Q \wedge R) \tag{7.6}$$

(iii) *Distributivity*

$$(P \vee Q) \wedge R = (P \wedge R) \vee (Q \wedge R)$$
$$(P \wedge Q) \vee R = (P \vee R) \wedge (Q \vee R) \tag{7.7}$$

(iv) *Identity*

$$P \vee \text{false} = P$$
$$P \wedge \text{True} = P$$
$$P \wedge \text{False} = \text{False}$$
$$P \vee \text{True} = \text{True} \tag{7.8}$$

(v) *Negation*

$$P \wedge \sim P = \text{False}$$
$$P \vee \sim P = \text{True} \tag{7.9}$$

(vi) *Idempotence*

$$P \vee P = P$$
$$P \wedge P = P \tag{7.10}$$

(vii) *Absorption*

$$P \wedge (P \vee Q) = P$$
$$P \vee (P \wedge Q) = P \tag{7.11}$$

(viii) *De Morgan's laws*

$$\sim(P \vee Q) = (\sim P \wedge \sim Q)$$
$$\sim(P \wedge Q) = (\sim P \vee \sim Q) \tag{7.12}$$

(ix) *Involution*

$$\sim(\sim P) = P \tag{7.13}$$

Each of these laws can be tested to be a tautology using truth tables.

Example 7.4

Verify De Morgan's laws.

(a) $\sim(P \vee Q) = (\sim P \wedge \sim Q)$

(b) $\sim(P \wedge Q) = (\sim P \vee \sim Q)$

Solution

(a)

P	Q	$P \vee Q$	A: $\sim(P \vee Q)$	$\sim P$	$\sim Q$	B: $\sim P \wedge \sim Q$	$A = B$
T	T	T	F	F	F	F	T
T	F	T	F	F	T	F	T
F	T	T	F	T	F	F	T
F	F	F	T	T	T	T	T

Therefore, $\sim(P \vee Q) = (\sim P \wedge \sim Q)$

(b)

P	Q	$P \wedge Q$	A: $\sim(P \wedge Q)$	$\sim P$	$\sim Q$	B: $\sim P \vee \sim Q$	$A = B$
T	T	T	F	F	F	F	T
T	F	F	T	F	T	T	T
F	T	F	T	T	T	T	T
F	F	F	T	T	F	T	T

Therefore $\sim(P \wedge Q) = (\sim P \vee \sim Q)$

Example 7.5

Simplify $(\sim(P \wedge Q) \Rightarrow R) \wedge P \wedge Q$

Solution

Consider $(\sim(P \wedge Q) \Rightarrow R) \wedge P \wedge Q$

$= (\sim \sim(P \wedge Q) \vee R) \wedge P \wedge Q$

(by eliminating '$\Rightarrow$' using $(P \Rightarrow Q) = (\sim P \vee Q)$)

$= ((P \wedge Q) \vee R) \wedge P \wedge Q$ (by the law of involution)

$= (P \wedge Q)$ (by the law of absorption)

7.1.2 Inference in Propositional Logic

Inference is a technique by which, given a set of *facts* or *postulates* or *axioms* or *premises* F_1, $F_2, \ldots, F_n$, a *goal* G is to be derived. For example, from the facts "Where there is smoke there is fire", and "There is smoke in the hill", the statement "Then the hill is on fire" can be easily deduced.

In propositional logic, three rules are widely used for inferring facts, namely

(i) *Modus Ponens*
(ii) *Modus Tollens*, and
(iii) *Chain rule*

Modus ponens (mod pons)

Given $P \Rightarrow Q$ and P to be true, Q is true.

$$\begin{array}{c} P \Rightarrow Q \\ P \\ \hline Q \end{array} \tag{7.14}$$

Here, the formulae above the line are the *premises* and the one below is the *goal* which can be inferred from the premises.

Modus tollens

Given $P \Rightarrow Q$ and $\sim Q$ to be true, $\sim P$ is true.

$$\begin{array}{c} P \Rightarrow Q \\ \sim Q \\ \hline \sim P \end{array} \tag{7.15}$$

Chain rule

Given $P \Rightarrow Q$ and $Q \Rightarrow R$ to be true, $P \Rightarrow R$ is true.

$$\begin{array}{c} P \Rightarrow Q \\ Q \Rightarrow R \\ \hline P \Rightarrow R \end{array} \tag{7.16}$$

Note that the chain rule is a representation of the *transitivity* relation with respect to the '$\Rightarrow$' connective.

Example 7.6

Given

(i) $C \vee D$
(ii) $\sim H \Rightarrow (A \wedge \sim B)$
(iii) $(C \vee D) \Rightarrow \sim H$
(iv) $(A \wedge \sim B) \Rightarrow (R \vee S)$

Can $(R \vee S)$ be inferred from the above?

Solution

From (i) and (iii) using the rule of Modus Ponens , $\sim H$ can be inferred.

$$\begin{array}{ll} \text{(i)} & C \vee D \\ \text{(iii)} & (C \vee D) \Rightarrow \sim H \\ \hline & \sim H \quad \text{(v)} \end{array}$$

From (ii) and (iv) using the chain rule, $\sim H \Rightarrow (R \vee S)$ can be inferred.

$$\begin{array}{ll} \text{(ii)} & \sim H \Rightarrow (A \wedge \sim B) \\ \text{(iv)} & (A \wedge \sim B) \Rightarrow (R \vee S) \\ \hline & \sim H \Rightarrow (R \vee S) \quad \text{(vi)} \end{array}$$

From (v) and (vi) using the rule of Modus Ponens $(R \vee S)$ can be inferred.

$$\begin{array}{ll} \text{(vi)} & \sim H \Rightarrow (R \vee S) \\ \text{(v)} & \sim H \\ \hline & R \vee S \end{array}$$

Hence, the result.

7.2 PREDICATE LOGIC

Defn

In propositional logic, events are symbolised as propositions which acquire either 'True/False' values. However, there are situations in the real world where propositional logic falls short of its expectation. For example, consider the following statements:

P : All men are mortal.

Q : Socrates is a man.

From the given statements it is possible to infer that Socrates is mortal. However, from the propositions P, Q which symbolise these statements nothing can be made out. The reason being, propositional logic lacks the ability to symbolise *quantification*. Thus, in this example, the quantifier "All" which represents the entire class of men encompasses Socrates as well, who is declared to be a man, in proposition Q. Therefore, by virtue of the first proposition P, Socrates who is a man also becomes a mortal, giving rise to the deduction Socrates is mortal. However, the deduction is not directly perceivable owing to the shortcomings in propositional logic. Therefore, propositional logic needs to be augmented with more tools to enhance its logical abilities.

Predicate logic comprises the following apart from the connectives and propositions recognized by propositional logic.

(i) Constants

(ii) Variables

(ii) Predicates

(iv) Quantifiers

(v) Functions

Constants represent objects that do not change values.

Example Pencil, Ram, Shaft, 100°C.

Variables are symbols which represent values acquired by the objects as qualified by the quantifier with which they are associated with.

Example x, y, z.

Predicates are representative of associations between objects that are constants or variables and acquire truth values 'True' or 'False'. A *predicate* carries a name representing the association followed by its arguments representing the objects it is to associate.

Example

likes (Ram, tea) (Ram likes tea)

plays (Sita, x) (Sita plays anything)

Here, likes and plays are predicate names and Ram, tea and Sita, x are the associated objects. Also, the predicates acquire truth values. If Ram disliked tea, likes (Ram, tea) acquires the value *false* and if Sita played any sport, plays (Sita, x) would acquire the value *true* provided x is suitably qualified by a quantifier.

Quantifiers are symbols which indicate the two types of quantification, namely, *All* ($\forall$) and *Some* ($\exists$). '$\forall$' is termed *universal quantifier* and '$\exists$' is termed *existential quantifier*.

Example Let,

man (x) : x is a man.
mortal (x) : x is mortal.
mushroom (x) : x is a mushroom.
poisonous (x) : x is poisonous.

Then, the statements

All men are mortal.
Some mushrooms are poisonous.

are represented as

$$\forall x\ (\text{man}\ (x) \Rightarrow \text{mortal}\ (x))$$

$$\exists x\ (\text{mushroom}\ (x) \wedge \text{poisonous}\ (x))$$

Here, a useful rule to follow is that a universal quantifier goes with implication and an existential quantifier with conjunction. Also, it is possible for logical formula to be quantified by multiple quantifiers.

Example Every ship has a captain.

$\forall x\ \exists y\ (\text{ship}\ (x) \Rightarrow \text{captain}\ (x, y))$

where, ship (x) : x is a ship

captain (x, y) : y is the captain of x.

Functions are similar to predicates in form and in their representation of association between objects but unlike predicates which acquire truth values alone, functions acquire values other than truth values. Thus, functions only serve as object descriptors.

Example

plus (2, 3) (2 plus 3 which is 5)

mother (Krishna) (Krishna's mother)

Observe that plus () and mother () indirectly describe "5" and "Krishna's mother" respectively.

Example 7.7

Write predicate logic statements for

(i) Ram likes all kinds of food.

(ii) Sita likes anything which Ram likes.

(iii) Raj likes those which Sita and Ram both like.

(iv) Ali likes some of which Ram likes.

Solution

Let food (x) : x is food.

likes (y, x) : y likes x

Then the above statements are translated as

(i) $\forall x$ food $(x) \Rightarrow$ likes (Ram, x))

(ii) $\forall x$ (likes (Ram, x) $\Rightarrow$ likes (Sita, x))

(iii) $\forall x$ (likes (Sita, x) $\wedge$ likes (Ram, x)) $\Rightarrow$ likes (Raj, x))

(iv) $\exists x$ (likes (Ram, x) $\wedge$ likes (Ali, x))

The application of the rule of universal quantifier and rule of existential quantifier can be observed in the translations given above.

7.2.1 Interpretations of Predicate Logic Formula

For a formula in propositional logic, depending on the truth values acquired by the propositions, the truth table interprets the formula. But in the case of predicate logic, depending on the truth values acquired by the predicates, the nature of the quantifiers, and the values taken by the constants and functions over a domain D, the formula is interpreted.

Example

Interpret the formulae

(i) $\forall x\ p(x)$

(ii) $\exists x\ p(x)$

where the domain $D = \{1, 2\}$ and

$p(1)$	$p(2)$
True	False

Solution

(i) $\forall x\, p(x)$ is true only if $p(x)$ is true for all values of x in the domain D, otherwise it is false. Here, for $x = 1$ and $x = 2$, the two possible values for x chosen from D, namely $p(1)$ = true and $p(2)$ = false respectively, yields (i) to be false since $p(x)$ is not true for $x = 2$. Hence, $\forall x\, p(x)$ is false.

(ii) $\exists x\, p(x)$ is true only if there is atleast one value of x for which $p(x)$ is true. Here, for $x = 1$, $p(x)$ is true resulting in (ii) to be true. Hence, $\exists x\, p(x)$ is true.

Example 7.8

Interpret $\forall x\, \exists y\, P(x, y)$ for $D = \{1, 2\}$ and

$P(1, 1)$	$P(1, 2)$	$P(2, 1)$	$P(2, 2)$
True	False	False	True

Solution

For $x = 1$, there exists a y, $(y = 1)$ for which $P(x, y)$, i.e. $(P(1,1))$ is true.

For $x = 2$, there exists a y, $(y = 2)$ for which $P(x, y)$ $(P(2, 2))$ is true.

Thus, for all values of x there exists a y for which $P(x, y)$ is true.

Hence, $\forall x\, \exists y\, P(x, y)$ is true.

7.2.2 Inference in Predicate Logic

The rules of inference such as Modus Ponens, Modus Tollens and Chain rule, and the laws of propositional logic are applicable for inferring predicate logic but not before the quantifiers have been appropriately eliminated (refer Chang & Lee, 1973).

Example

Given (i) All men are mortal.

(ii) Confucius is a man.

Prove: Confucius is mortal.

Translating the above into predicate logic statements

(i) $\forall x\, (\text{man}\ (x) \Rightarrow \text{mortal}\ (x))$

(ii) man (Confucius)

(iii) mortal (Confucius)

Since (i) is a tautology qualified by the universal quantifier for x = Confucius, the statement is true, i.e.

$$\text{man (Confucius)} \Rightarrow \text{mortal (Confucius)}$$
$$\Rightarrow\ \sim\text{man (Confucius)} \vee \text{mortal (Confucius)}$$

But from (ii), man (Confucius) is true.
Hence (iv) simplifies to

$$\text{False} \vee \text{mortal (Confucius)}$$
$$= \text{mortal (Confucius)}$$

Hence, Confucius is mortal has been proved.

Example 7.9

Given (i) Every soldier is strong-willed.

(ii) All who are strong-willed and sincere will succeed in their career.

(iii) Indira is a soldier.

(iv) Indira is sincere.

Prove: Will Indira succeed in her career?

Solution

Let soldier (x) : x is a soldier.
strong-willed (x) : x is a strong-willed.
sincere (x) : x is sincere.
succeed_career (x) : x succeeds in career.

Now (i) to (iv) are translated as

$$\forall x\ (\text{soldier}\ (x) \Rightarrow \text{strong-willed}\ (x)) \quad \text{(i)}$$

$$\forall x\ ((\text{strong-willed}\ (x) \wedge \text{sincere}\ (x)) \Rightarrow \text{succeed\_career}\ (x)) \quad \text{(ii)}$$

$$\text{soldier (Indira)} \quad \text{(iii)}$$

$$\text{sincere (Indira)} \quad \text{(iv)}$$

To show whether Indira will succeed in her career, we need to show

$$\text{succeed\_career(Indira) is true.} \quad \text{(v)}$$

Since (i) and (ii) are quantified by $\forall$, they should be true for x = Indira.
Substituting x = Indira in (i) results in (soldier (Indira) $\Rightarrow$ strong-willed (Indira),

i.e.
$$\sim\text{soldier (Indira)} \vee \text{strong-willed (Indira)} \quad \text{(vi)}$$

Since from (iii) soldier (Indira) is true, (vi) simplifies to

$$\text{strong-willed (Indira)} \quad \text{(vii)}$$

Substituting x = Indira in (ii),

$$(\text{strong-willed (Indira)} \wedge \text{sincere (Indira)}) \Rightarrow \text{succeed\_career (Indira)}$$

i.e.
$$\sim(\text{strong-willed (Indira)} \wedge \text{sincere (Indira)}) \vee \text{succeed\_career (Indira)}$$
$$(\Theta\ P \Rightarrow Q = \sim P \vee Q))$$

i.e.
$$\sim(\text{strong-willed (Indira)} \vee \sim\text{sincere (Indira)}) \vee \text{succeed\_career (Indira)}$$
$$\text{(De Morgan's law) (viii)}$$

From (vii), strong-willed (Indira) is true and from (iv) sincere (Indira) is true. Substituting these in (viii),

$$\text{False} \vee \text{False} \vee \text{succeed\_career (Indira)}$$

i.e.
$$\text{succeed\_career (Indira)} \quad \text{(using law of identity)}$$

Hence, Indira will succeed in her career is true.

7.3 FUZZY LOGIC

In crisp logic, the truth values acquired by propositions or predicates are 2-valued, namely *True*, *False* which may be treated numerically equivalent to (0, 1). However, in fuzzy logic, truth values are multivalued such as *absolutely true, partly true, absolutely false, very true,* and so on and are numerically equivalent to (0–1).

Fuzzy propositions

A *fuzzy proposition* is a statement which acquires a fuzzy truth value. Thus, given $\tilde{P}$ to be a fuzzy proposition, $T(\tilde{P})$ represents the truth value (0–1) attached to $\tilde{P}$. In its simplest form, fuzzy propositions are associated with fuzzy sets. The fuzzy membership value associated with the fuzzy set $\tilde{A}$ for $\tilde{P}$ is treated as the fuzzy truth value $T(\tilde{P})$.

i.e. $T(\tilde{P}) = \mu_{\tilde{A}}(x)$ where $0 \le \mu_{\tilde{A}}(x) \le 1$ (7.17)

Example

$\tilde{P}$: Ram is honest.

$T(\tilde{P}) = 0.8$, if $\tilde{P}$ is partly true.

$T(\tilde{P}) = 1$, if $\tilde{P}$ is absolutely true.

Fuzzy connectives

Fuzzy logic similar to crisp logic supports the following connectives:

(i) *Negation* : $-$

(ii) *Disjunction* : $\vee$

(iii) *Conjunction* : $\wedge$

(iv) *Implication* : $\Rightarrow$

Table 7.3 illustrates the definition of the connectives. Here $\tilde{P}$, $\tilde{Q}$ are fuzzy propositions and $T(\tilde{P})$, $T(\tilde{Q})$, are their truth values.

Table 7.3 Fuzzy connectives

Symbol	Connective	Usage	Definition
$-$	Negation	$\bar{\tilde{P}}$	$1 - T(\tilde{P})$
$\vee$	Disjunction	$\tilde{P} \vee \tilde{Q}$	$\max(T(\tilde{P}), T(\tilde{Q}))$
$\wedge$	Conjunction	$\tilde{P} \wedge \tilde{Q}$	$\min(T(\tilde{P}), T(\tilde{Q}))$
$\Rightarrow$	Implication	$\tilde{P} \Rightarrow \tilde{Q}$	$\bar{\tilde{P}} \vee \tilde{Q} = \max(1 - T(\tilde{P}), T(\tilde{Q}))$

$\tilde{P}$ and $\tilde{Q}$ related by the '$\Rightarrow$' operator are known as antecedent and consequent respectively. Also, just as in crisp logic, here too, '$\Rightarrow$' represents the IF-THEN statement as

IF x is $\tilde{A}$ THEN y is $\tilde{B}$, and is equivalent to

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\bar{\tilde{A}} \times Y) \tag{7.18}$$

The membership function of $\tilde{R}$ is given by

$$\mu_{\tilde{R}}(x, y) = \max(\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), 1 - \mu_{\tilde{A}}(x)) \tag{7.19}$$

Also, for the compound implication IF x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is $\tilde{C}$ the relation R is equivalent to

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\bar{\tilde{A}} \times \tilde{C}) \tag{7.20}$$

The membership function of $\tilde{R}$ is given by

$$\mu_{\tilde{R}}(x, y) = \max(\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), \min(1 - \mu_{\tilde{A}}(x), \mu_{\tilde{C}}(y))) \tag{7.21}$$

Example

$\tilde{P}$: Mary is efficient, $T(\tilde{P}) = 0.8$

$\tilde{Q}$: Ram is efficient, $T(\tilde{Q}) = 0.65$

(i) $\bar{\tilde{P}}$: Mary is not efficient.

$$T(\bar{\tilde{P}}) = 1 - T(\tilde{P}) = 1 - 0.8 = 0.2$$

(ii) $\tilde{P} \wedge \tilde{Q}$: Mary is efficient and so is Ram.

$$\begin{aligned} T(\tilde{P} \wedge \tilde{Q}) &= \min(T(\tilde{P}), T(\tilde{Q})) \\ &= \min(0.8, 0.65) \\ &= 0.65 \end{aligned}$$

(iii) $T(\tilde{P} \vee \tilde{Q})$: Either Mary or Ram is efficient.

$$\begin{aligned} T(\tilde{P} \vee \tilde{Q}) &= \max(T(\tilde{P}), T(\tilde{Q})) \\ &= \max(0.8, 0.65) \\ &= 0.8 \end{aligned}$$

(iv) $\tilde{P} \Rightarrow \tilde{Q}$: If Mary is efficient then so is Ram.

$$\begin{aligned} T(\tilde{P} \Rightarrow \tilde{Q}) &= \max(1 - T(\tilde{P}), T(\tilde{Q})) \\ &= \max(0.2, 0.65) \\ &= 0.65 \end{aligned}$$

Example 7.10

Let $X = \{a, b, c, d\}$ $Y = \{1, 2, 3, 4\}$

and $\tilde{A} = \{(a, 0)(b, 0.8)(c, 0.6)(d, 1)\}$

$\tilde{B} = \{(1, 0.2)(2, 1)(3, 0.8)(4, 0)\}$

$\tilde{C} = \{(1, 0)(2, 0.4)(3, 1)(4, 0.8)\}$

1 0.2 0.4 0
(a, 1) (b, 0.2) (c 0.4) 0.2 d, 0)
0.3
1 0.5

Determine the implication relations

(i) IF x is $\tilde{A}$ THEN y is $\tilde{B}$.

(ii) IF x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is $\tilde{C}$.

Solution

To determine (i) compute

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\bar{\tilde{A}} \times Y) \quad \text{where}$$

$$\mu_{\tilde{R}}(x, y) = \max(\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), 1 - \mu_{\tilde{A}}(x))$$

$$\tilde{A} \times \tilde{B} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0.8 & 0 \\ 0.2 & 0.6 & 0.6 & 0 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

$$\bar{\tilde{A}} \times Y = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.4 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Here, Y the universe of discourse could be viewed as $\{(1, 1)\ (2, 1)\ (3, 1)\ (4, 1)\}$ a fuzzy set all of whose elements x have $\mu(x) = 1$.

Therefore,

$$\tilde{R} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 0.8 & 0.8 & 0.2 \\ 0.4 & 0.6 & 0.6 & 0.4 \\ 0.2 & 0.1 & 0.8 & 0 \end{bmatrix} \end{array}$$

which represents IF x is $\tilde{A}$ THEN y is $\tilde{B}$.

To determine (ii) compute

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\bar{\tilde{A}} \times \tilde{C}) \text{ where}$$

$$\mu_{\tilde{R}}(x, y) = \max\left(\min\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\right), \min\left(1 - \mu_{\tilde{A}}(x), \mu_{\tilde{C}}(y)\right)\right)$$

$$\tilde{A} \times \tilde{B} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0.8 & 0 \\ 0.2 & 0.6 & 0.6 & 0 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

$$\bar{\tilde{A}} \times \tilde{C} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0.4 & 1 & 0.8 \\ 0 & 0.2 & 0.2 & 0.2 \\ 0 & 0.4 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Therefore,

$$\tilde{R} = \max\left((\tilde{A} \times \tilde{B}), (\bar{\tilde{A}} \times \tilde{C})\right) \text{ gives}$$

$$\tilde{R} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0.4 & 1 & 0.8 \\ 0.2 & 0.8 & 0.8 & 0.2 \\ 0.2 & 0.6 & 0.6 & 0.4 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

The above $\tilde{R}$ represents IF x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is $\tilde{C}$.

7.3.1 Fuzzy Quantifiers

Just as in crisp logic where predicates are quantified by quantifiers, fuzzy logic propositions are also quantified by *fuzzy quantifiers*. There are two classes of fuzzy quantifiers such as

(i) Absolute quantifiers and

(ii) Relative quantifiers

While absolute quantifiers are defined over $\Re$, relative quantifiers are defined over [0–1].

13.2 LAMBDA-CUTS FOR FUZZY SETS (ALPHA-CUTS)

Consider a fuzzy set A. The set $A_\lambda (0 < \lambda < 1)$, called the lambda ($\lambda$)-cut (or alpha [$\alpha$]-cut) set, is a crisp set of the fuzzy set and is defined as follows:

$$A_\lambda = \{x \mid \mu_A(x) \geq \lambda\}; \quad \lambda \in [0,1]$$

The set A_λ is called a weak lambda cut set if it consists of all the elements of a fuzzy set whose membership functions have values greater than or equal to a specified value. On the other hand, the set A_λ is called a strong lambda-cut set if it consists of all the elements of a fuzzy set whose membership functions have values strictly greater than a specified value. A strong λ cut set is given by

$$A_\lambda = \{x \mid \mu_A(x) > \lambda\}; \quad \lambda \in [0,1]$$

5. Determine the crisp λ-cut relation when $\lambda = 0.1, 0^+$, 0.3 and 0.9 for the following relation $\underset{\sim}{R}$:

$$\underset{\sim}{R} = \begin{bmatrix} 0 & 0.2 & 0.4 \\ 0.3 & 0.7 & 0.1 \\ 0.8 & 0.9 & 1.0 \end{bmatrix}$$

Solution: For the given fuzzy relation, the λ-cut relation is given by

$$R_\lambda = \left\{(x, y) \middle| \mu_{\underset{\sim}{R}(x,y)} \geq \lambda\right\}$$

$$= \left\{1 \middle| \mu_{\underset{\sim}{R}(x,y)} \geq \lambda; 0 \middle| \mu_{\underset{\sim}{R}(x,y)} < \lambda\right\}$$

(a) $\lambda = 0.1$,

$$R_{0.1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(b) $\lambda = 0^+$,

$$R_{0^+} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(c) $\lambda = 0.3$,

$$R_{0.3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(d) $\lambda = 0.9$,

$$R_{0.9} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Genetic algorithm

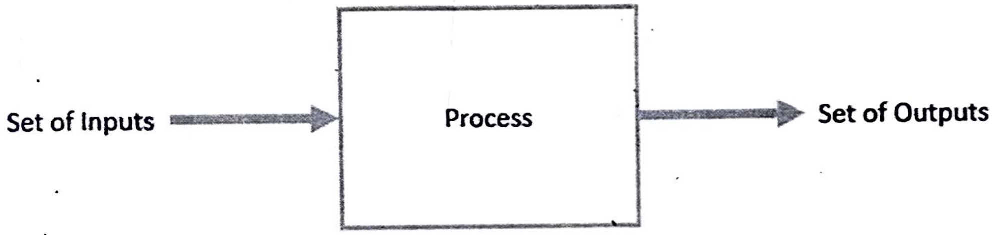
→ Genetic Algorithm (GA) is a search-based optimization technique based on the principles of **Genetics and Natural Selection**. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.

→ Provides efficient, effective technique for optimisation & ML applications.

→ Widely used in business, scientific & engineering circles.

Introduction to Optimization

Optimization is the process of **making something better**. In any process, we have a set of inputs and a set of outputs as shown in the following figure.



Optimization refers to finding the values of inputs in such a way that we get the "best" output values. The definition of "best" varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.

The set of all possible solutions or values which the inputs can take make up the search space. In this search space, lies a point or a set of points which gives the optimal solution. The aim of optimization is to find that point or set of points in the search space

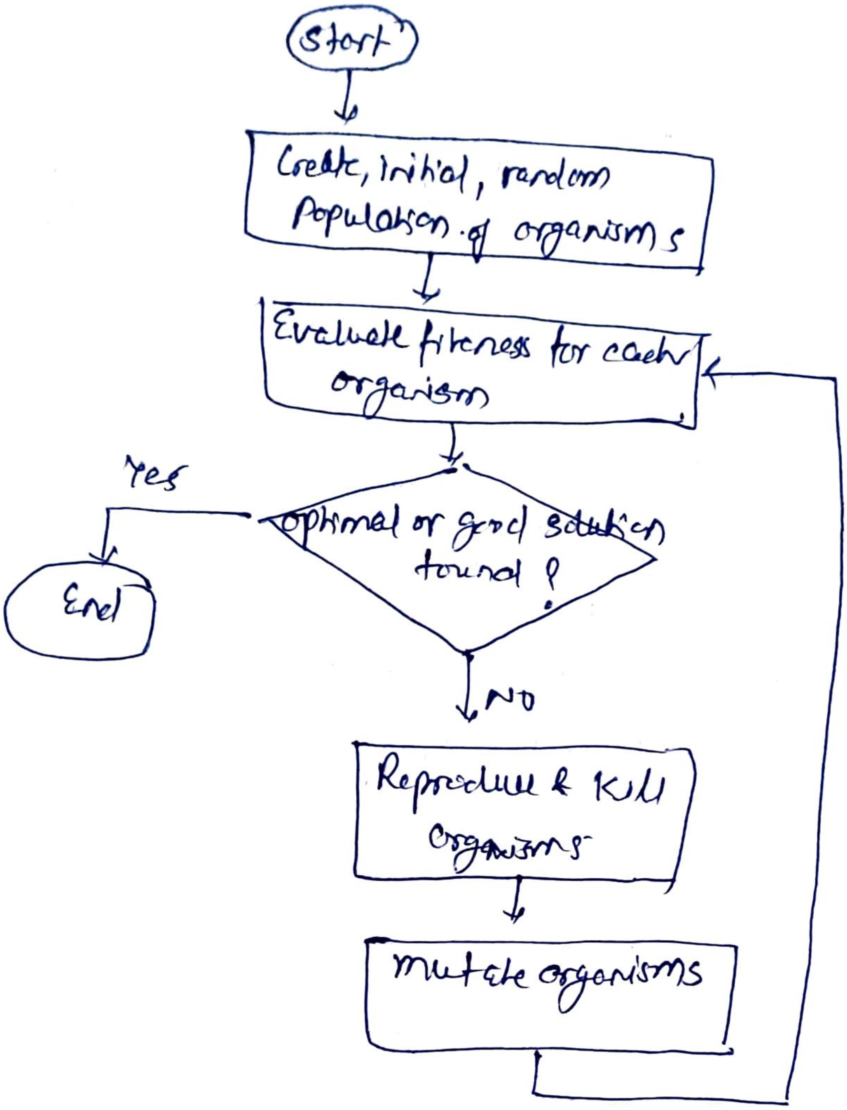
What are Genetic Algorithms?

Nature has always been a great source of inspiration to all mankind. Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics. GAs are a subset of a much larger branch of computation known as **Evolutionary Computation**.

GAs were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success.

In GAs, we have a **pool or a population of possible solutions** to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations. Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more "fitter" individuals. This is in line with the Darwinian Theory of "Survival of the Fittest".

Flow chart-



Genetic operators

1. Selection Operator
2. mutation
3. Crossover.

Selection operator

- Process of selecting two or more parents from the population
- Purpose of selection is to emphasize filter individuals in the population in hopes that their offsprings have higher fitness.

Que) Maximize the function $f(x) = x^2$ with x in interval $[0, 31]$.

1. Generate initial population as random they are chromosomes or genotypes.

   eg: 01101 (13), 11000 (24), 01000 (8), 10011 (19)

2. Calculate fitness

   a) Decode into an integer (called phenotypes)

   01101 → 13, 11000 → 24, 01000 → 8, 10011 → 19

   b) Evaluate fitness $f(x) = x^2$

   13 → 169, 24 → 576, 8 → 64, 19 → 361

$$P_i = f_i / \sum_{j=1}^{n} f_j$$

f_i = fitness for string i in populatn

P_i = Prob of string i being selected

n = no of Individual in the population

expected Count $= nP_i$

String no	Initial Populatn	xvalue	Fitness F_i $f(x) = x^2$	P_i	expected count nP_i
1	01101	13	169	0.14	0.58
2	11000	24	576	0.49	1.97
3	01000	8	64	0.06	0.22
4	10011	19	361	0.31	1.23
Sum			1170		

String No	mating Pool	crossover point	offspring after crossover	x value	fitness $f(x)=x^2$
1	01101	4	01100	12	144
2	11000	4	11001	25	625
2	11000	2	11011	27	729
4	10011	2	10000	16	256
					1754

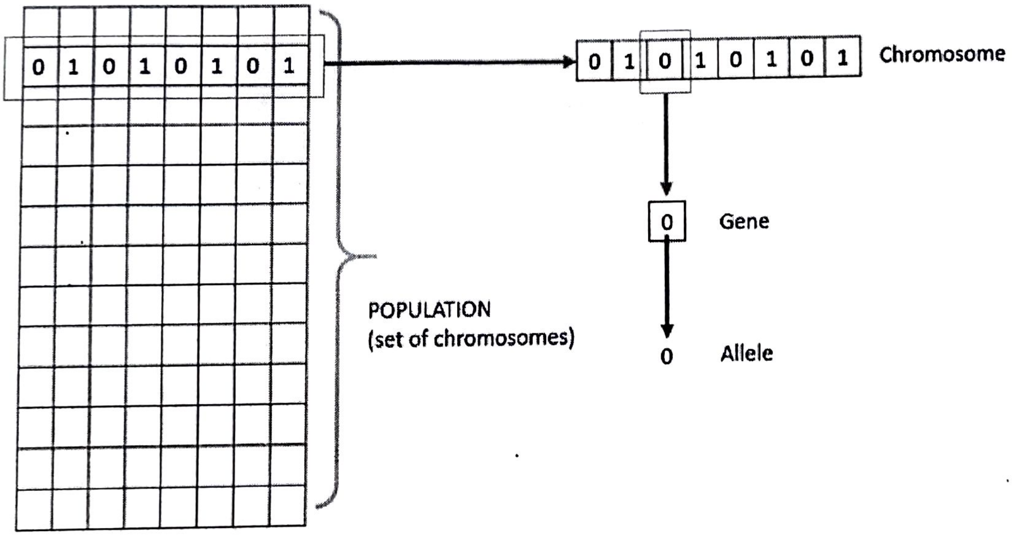
not apply mutation due to wrong fitness

String No	offspring after crossover	offspring after mutation	x value	fitness value $f(x)=x^2$
1	01100	11100	26	676
2	110001	11001	25	625
3	11011	11011	27	729
4	10000	10100	18	324
				2354

Basic Terminology.

Before beginning a discussion on Genetic Algorithms, it is essential to be familiar with some basic terminology which will be used throughout this tutorial.

- **Population** – It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings.
- **Chromosomes** – A chromosome is one such solution to the given problem.
- **Gene** – A gene is one element position of a chromosome.
- **Allele** – It is the value a gene takes for a particular chromosome.



- **Genotype** – Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.
- **Phenotype** – Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.
- **Decoding and Encoding** – For simple problems, the **phenotype and genotype** spaces are the same. However, in most of the cases, the phenotype and genotype spaces are different. Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. Decoding should be fast as it is carried out repeatedly in a GA during the fitness value calculation.

  For example, consider the 0/1 Knapsack Problem. The Phenotype space consists of solutions which just contain the item numbers of the items to be picked.

  However, in the genotype space it can be represented as a binary string of length n (where n is the number of items). A **0 at position x** represents that x^{th} item is picked

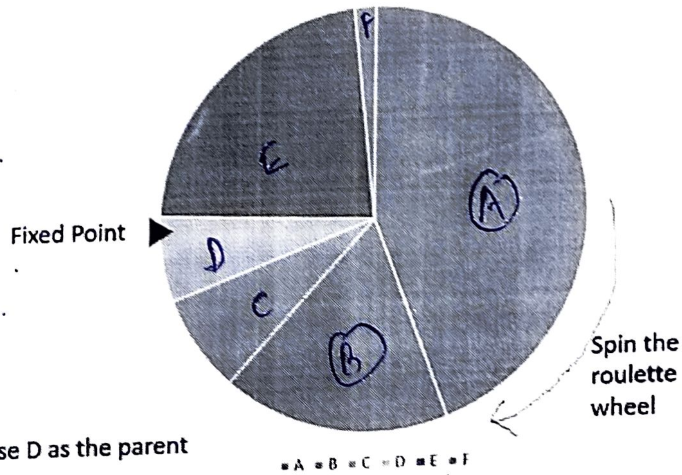
Consider a circular wheel. The wheel is divided into **n pies**, where n is the number of individuals in the population. Each individual gets a portion of the circle which is proportional to its fitness value.

Different selection methods are

Two implementations of fitness proportionate selection are possible –

1 Roulette Wheel Selection

In a roulette wheel selection, the circular wheel is divided as described before. A fixed point is chosen on the wheel circumference as shown and the wheel is rotated. The region of the wheel which comes in front of the fixed point is chosen as the parent. For the second parent, the same process is repeated.



Choose D as the parent

Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

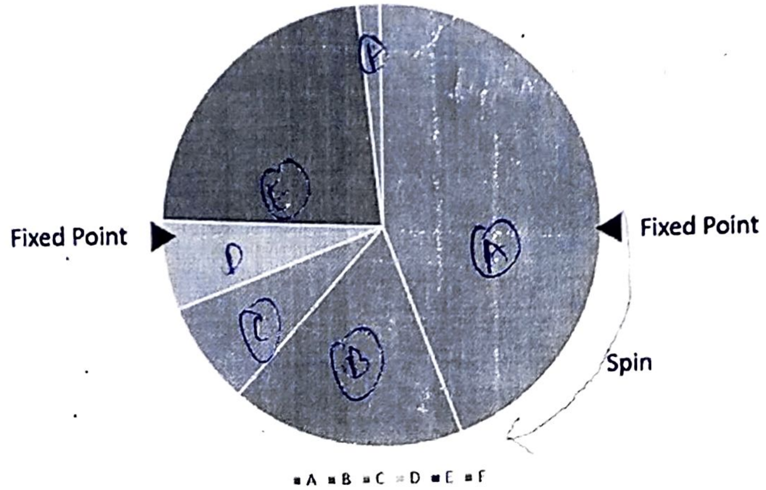
It is clear that a fitter individual has a greater pie on the wheel and therefore a greater chance of landing in front of the fixed point when the wheel is rotated. Therefore, the probability of choosing an individual depends directly on its fitness.

Implementation wise, we use the following steps –

- Calculate S = the sum of a finesses.
- Generate a random number between 0 and S.
- Starting from the top of the population, keep adding the finesses to the partial sum P, till P<S.
- The individual for which P exceeds S is the chosen individual.

2 Stochastic Universal Sampling (SUS)

Stochastic Universal Sampling is quite similar to Roulette wheel selection, however instead of having just one fixed point, we have multiple fixed points as shown in the following image. Therefore, all the parents are chosen in just one spin of the wheel. Also, such a setup encourages the highly fit individuals to be chosen at least once.

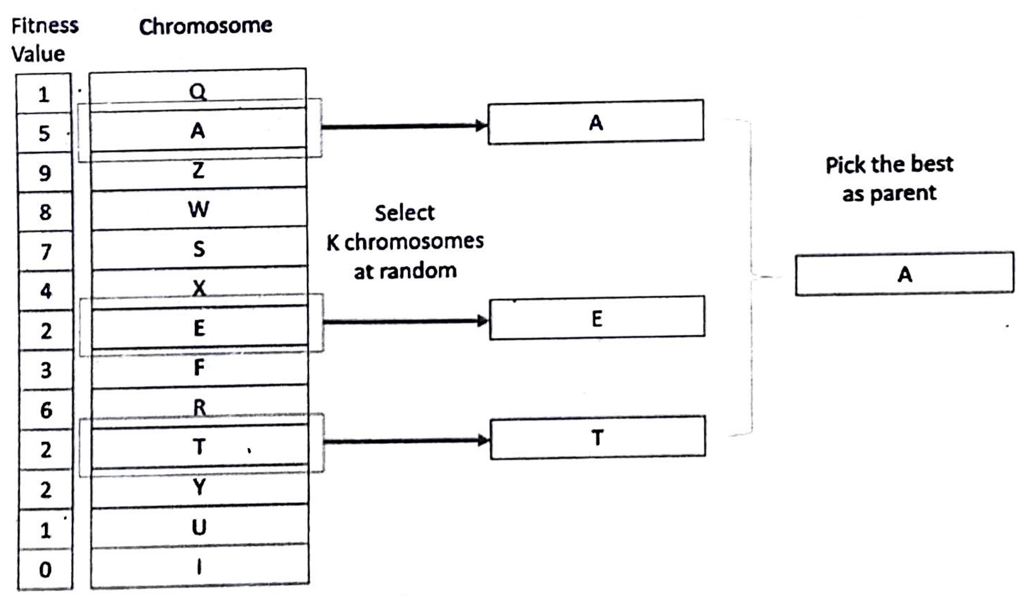


Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

It is to be noted that fitness proportionate selection methods don't work for cases where the fitness can take a negative value.

Tournament Selection

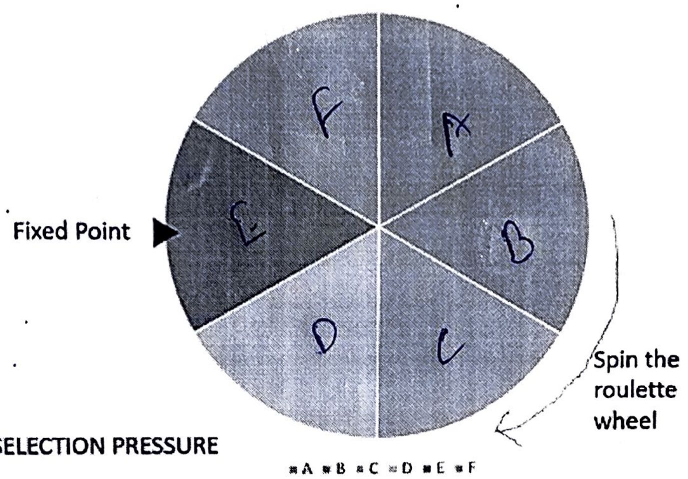
In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.



7

4 Rank Selection

Rank Selection also works with negative fitness values and is mostly used when the individuals in the population have very close fitness values (this happens usually at the end of the run). This leads to each individual having an almost equal share of the pie (like in case of fitness proportionate selection) as shown in the following image and hence each individual no matter how fit relative to each other has an approximately same probability of getting selected as a parent. This in turn leads to a loss in the selection pressure towards fitter individuals, making the GA to make poor parent selections in such situations.



Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

NO SELECTION PRESSURE

In this, we remove the concept of a fitness value while selecting a parent. However, every individual in the population is ranked according to their fitness. The selection of the parents depends on the rank of each individual and not the fitness. The higher ranked individuals are preferred more than the lower ranked ones.

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

Random Selection

In this strategy we randomly select parents from the existing population. There is no selection pressure towards fitter individuals and therefore this strategy is usually avoided.

Introduction to Crossover

The crossover operator is analogous to reproduction and biological crossover. In this more than one parent is selected and one or more off-springs are produced using the genetic material of the parents. Crossover is usually applied in a GA with a high probability – p_c .

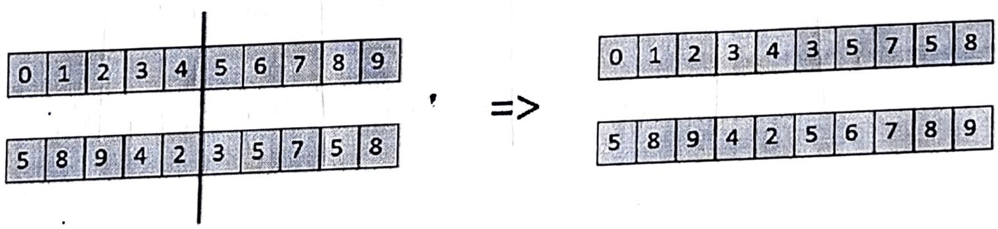
→ Can be one point or two point crossover

Crossover Operators

In this section we will discuss some of the most popularly used crossover operators. It is to be noted that these crossover operators are very generic and the GA Designer might choose to implement a problem-specific crossover operator as well.

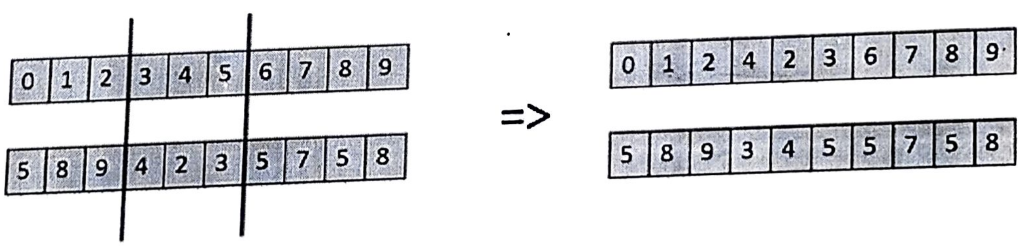
One Point Crossover

In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.



Multi Point Crossover

Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



Uniform Crossover

In a uniform crossover, we don't divide the chromosome into segments, rather we treat each gene separately. In this, we essentially flip a coin for each chromosome to decide whether or not

There exist a lot of other crossovers like Partially Mapped Crossover (PMX), Order based crossover (OX2), Shuffle Crossover, Ring Crossover, etc.

Introduction to Mutation

In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – p_m. If the probability is very high, the GA gets reduced to a random search.

Mutation is the part of the GA which is related to the "exploration" of the search space. It has been observed that mutation is essential to the convergence of the GA while crossover is not.

→ Applied to each child individually after crossover
→ Bits are changed from 0 to 1 or from 1 to 0 randomly chosen position of randomly selected string.

Mutation Operators

In this section, we describe some of the most commonly used mutation operators. Like the crossover operators, this is not an exhaustive list and the GA designer might find a combination of these approaches or a problem-specific mutation operator more useful.

Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

0	0	1	1	0	1	0	0	1	0

=>

0	0	1	0	0	1	0	0	1	0

Random Resetting

Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.

1	2	3	4	5	6	7	8	9	0

=>

1	6	3	4	5	2	7	8	9	0

Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.

0	1	2	3	4	5	6	7	8	9

=>

0	1	3	6	4	2	5	7	8	9

Inversion Mutation

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.

0	1	2	3	4	5	6	7	8	9

=>

0	1	6	5	4	3	2	7	8	9

The Survivor Selection Policy determines which individuals are to be kicked out and which are to be kept in the next generation. It is crucial as it should ensure that the fitter individuals are not kicked out of the population, while at the same time diversity should be maintained in the population.

Some GAs employ **Elitism**. In simple terms, it means the current fittest member of the population is always propagated to the next generation. Therefore, under no circumstance can the fittest member of the current population be replaced.

The easiest policy is to kick random members out of the population, but such an approach frequently has convergence issues, therefore the following strategies are widely used.

Age Based Selection

In Age-Based Selection, we don't have a notion of a fitness. It is based on the premise that each individual is allowed in the population for a finite generation where it is allowed to reproduce, after that, it is kicked out of the population no matter how good its fitness is.

For instance, in the following example, the age is the number of generations for which the individual has been in the population. The oldest members of the population i.e. P4 and P7 are kicked out of the population and the ages of the rest of the members are incremented by one.

Age	Fitness Value	Chromosome
6	1	P1
8	5	P2
5	9	P3
10	8	P4
2	7	P5
3	4	P6
9	2	P7
2	2	P8
5	1	P9
4	0	P10

EXISTING POPULATION

+

Fitness Value	Chromosome
8	C1
9	C2

OFFSPRINGS

=>

Age	Fitness Value	Chromosome
7	1	P1
9	5	P2
4	9	P3
0	8	C1
3	7	P5
4	4	P6
0	9	C2
3	2	P8
6	1	P9
5	0	P10

NEW POPULATION

Questions 15: Genetic Algorithms

Roman Belavkin

Middlesex University

Question 1

Give an example of combinatorial problem. What is the most difficult in solving these problems?

Answer: *One classical example is the Travelling Salesman problem (TSP), described in the lecture notes. Another example is the timetable problem. The main difficulty is that the number of combinations (and, hence, the number of possible solutions) grows much faster than the number of items involved in the problem (i.e. the number of cities in TSP, the number of time–slots, etc). This problem is known as combinatorial explosion.*

Question 2

Name and describe the main features of Genetic Algorithms (GA).

Answer: *Genetic Algorithms (GA) use principles of natural evolution. There are five important features of GA:*

Encoding *possible solutions of a problem are considered as individuals in a population. If the solutions can be divided into a series of small steps (building blocks), then these steps are represented by genes and a series of genes (a chromosome) will encode the whole solution. This way different solutions of a problem are represented in GA as chromosomes of individuals.*

Fitness Function *represents the main requirements of the desired solution of a problem (i.e. cheapest price, shortest route, most compact arrangement, etc). This function calculates and returns the fitness of an individual solution.*

Selection *operator defines the way individuals in the current population are selected for reproduction. There are many strategies for that (e.g. roulette–wheel, ranked, tournament selection, etc), but usually the individuals which are more fit are selected.*

Crossover *operator defines how chromosomes of parents are mixed in order to obtain genetic codes of their offspring (e.g. one–point, two–point, uniform crossover, etc). This operator implements the inheritance property (offspring inherit genes of their parents).*

Mutation *operator creates random changes in genetic codes of the offspring. This operator is needed to bring some random diversity into the genetic code. In some cases GA cannot find the optimal solution without mutation operator (local maximum problem).*

Question 3

Consider the problem of finding the shortest route through several cities, such that each city is visited only once and in the end return to the starting city (the Travelling Salesman problem). Suppose that in order to solve this problem we use a genetic algorithm, in which genes represent links between pairs of cities. For example, a link between London and Paris is represented by a single gene 'LP'. Let also assume that the direction in which we travel is not important, so that $LP = PL$.

a) How many genes will be used in a chromosome of each individual if the number of cities is 10?

Answer: *Each chromosome will consist of 10 genes. Each gene representing the path between a pair of cities in the tour.*

b) How many genes will be in the alphabet of the algorithm?

Answer: *The alphabet will consist of 45 genes. Indeed, each of the 10 cities can be connected with 9 remaining. Thus,* $10 \times 9 = 90$ *is the number of ways in which 10 cities can be grouped in pairs. However, because the direction is not important (i.e. London–Paris is the same as Paris–London) the number must be divided by 2. So, we shall need* $90/2 = 45$ *genes in order to encode all pairs. In general the formula for* n *cities is:*

$$\frac{n(n-1)}{2}$$

Question 4

Suppose a genetic algorithm uses chromosomes of the form $x = abcdefgh$ with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as:

$$f(x) = (a+b) - (c+d) + (e+f) - (g+h) ,$$

and let the initial population consist of four individuals with the following chromosomes:

$$\begin{aligned} x_1 &= 6\,5\,4\,1\,3\,5\,3\,2 \\ x_2 &= 8\,7\,1\,2\,6\,6\,0\,1 \\ x_3 &= 2\,3\,9\,2\,1\,2\,8\,5 \\ x_4 &= 4\,1\,8\,5\,2\,0\,9\,4 \end{aligned}$$

a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.

Answer:

$$\begin{aligned} f(x_1) &= (6+5)-(4+1)+(3+5)-(3+2) = 9 \\ f(x_2) &= (8+7)-(1+2)+(6+6)-(0+1) = 23 \\ f(x_3) &= (2+3)-(9+2)+(1+2)-(8+5) = -16 \\ f(x_4) &= (4+1)-(8+5)+(2+0)-(9+4) = -19 \end{aligned}$$

The order is x_2, x_1, x_3 *and* x_4.

b) Perform the following crossover operations:

i) Cross the fittest two individuals using one–point crossover at the middle point.

Answer: *One–point crossover on* x_2 *and* x_1:

$$\begin{array}{rr|r} x_2 = & \mathbf{8\,7\,1\,2} & \mathbf{6\,6\,0\,1} \\ x_1 = & 6\,5\,4\,1 & 3\,5\,3\,2 \end{array} \Rightarrow \begin{array}{rl} O_1 = & \mathbf{8\,7\,1\,2}\,3\,5\,3\,2 \\ O_2 = & 6\,5\,4\,1\,\mathbf{6\,6\,0\,1} \end{array}$$

ii) Cross the second and third fittest individuals using a two–point crossover (points b and f).

Answer: *Two–point crossover on* x_1 *and* x_3

$$\begin{array}{rr|r|r} x_1 = & \mathbf{6\,5} & \mathbf{4\,1\,3\,5} & \mathbf{3\,2} \\ x_3 = & 2\,3 & 9\,2\,1\,2 & 8\,5 \end{array} \Rightarrow \begin{array}{rl} O_3 = & \mathbf{6\,5}\,9\,2\,1\,2\,\mathbf{3\,2} \\ O_4 = & 2\,3\,\mathbf{4\,1\,3\,5}\,8\,5 \end{array}$$

iii) Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.

Answer: *In the simplest case uniform crossover means just a random exchange of genes between two parents. For example, we may swap genes at positions a, d and f of parents x_2 and x_3:*

$$\begin{array}{ll} x_2 = \underline{8}\,7\,1\,\underline{2}\,6\,\underline{6}\,0\,1 \\ x_3 = \underline{2}\,3\,9\,\underline{2}\,1\,\underline{2}\,8\,5 \end{array} \Rightarrow \begin{array}{ll} O_5 = 2\,7\,1\,2\,6\,2\,0\,1 \\ O_6 = 8\,3\,9\,2\,1\,6\,8\,5 \end{array}$$

c) Suppose the new population consists of the six offspring individuals received by the crossover operations in the above question. Evaluate the fitness of the new population, showing all your workings. Has the overall fitness improved?

Answer: *The new population is:*

$$\begin{aligned} O_1 &= 8\,7\,1\,2\,3\,5\,3\,2 \\ O_2 &= 6\,5\,4\,1\,6\,6\,0\,1 \\ O_3 &= 6\,5\,9\,2\,1\,2\,3\,2 \\ O_4 &= 2\,3\,4\,1\,3\,5\,8\,5 \\ O_5 &= 2\,7\,1\,2\,6\,2\,0\,1 \\ O_6 &= 8\,3\,9\,2\,1\,6\,8\,5 \end{aligned}$$

Now apply the fitness function $f(x) = (a+b)-(c+d)+(e+f)-(g+h)$:

$$\begin{aligned} f(O_1) &= (8+7)-(1+2)+(3+5)-(3+2) = 15 \\ f(O_2) &= (6+5)-(4+1)+(6+6)-(0+1) = 17 \\ f(O_3) &= (6+5)-(9+2)+(1+2)-(3+2) = -2 \\ f(O_4) &= (2+3)-(4+1)+(3+5)-(8+5) = -5 \\ f(O_5) &= (2+7)-(1+2)+(6+2)-(0+1) = 13 \\ f(O_6) &= (8+3)-(9+2)+(1+6)-(8+5) = -6 \end{aligned}$$

The overall fitness has improved.

d) By looking at the fitness function and considering that genes can only be digits between 0 and 9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of the maximum fitness.

Answer: *The optimal solution should have a chromosome that gives the maximum of the fitness function*

$$\max f(x) = \max\left[(a+b)-(c+d)+(e+f)-(g+h)\right] .$$

Because genes can only be digits from 0 to 9, the optimal solution should be:

$$x_{\text{optimal}} = 9\,9\,0\,0\,9\,9\,0\,0\,,$$

and the maximum fitness is

$$f(x_{\text{optimal}}) = (9+9) - (0+0) + (9+9) - (0+0) = 36$$

e) By looking at the initial population of the algorithm can you say whether it will be able to reach the optimal solution without the mutation operator?

Answer: *No, the algorithm will never reach the optimal solution without mutation. The optimal solution is* $x_{\text{optimal}} = 9\,9\,0\,0\,9\,9\,0\,0$*. If mutation does not occur, then the only way to change genes is by applying the crossover operator. Regardless of the way crossover is performed, its only outcome is an exchange of genes of parents at certain positions in the chromosome. This means that the first gene in the chromosomes of children can only be either 6, 8, 2 or 4 (i.e. first genes of* x_1, x_2, x_3 *and* x_4*), and because none of the individuals in the initial population begins with gene 9, the crossover operator alone will never be able to produce an offspring with gene 9 in the beginning. One can easily check that a similar problem is present at several other positions. Thus, without mutation, this GA will not be able to reach the optimal solution.*

Question 5

What two requirements should a problem satisfy in order to be suitable for solving it by a GA?

Answer: *GA can only be applied to problems that satisfy the following requirements:*

- *The fitness function can be well–defined.*
- *Solutions should be decomposable into steps (building blocks) which could be then encoded as chromosomes.*

Question 6

A budget ariline company operates 3 plains and employs 5 cabin crews. Only one crew can operate on any plain on a single day, and each crew cannot work for more than two days in a row. The company uses all planes every day. A Genetic Algorithm is used to work out the best combination of crews on any particular day.

a) Suggest what chromosome could represent an individual in this algorithm?

Answer: *On each day, a solution is a combination of 3 cabin crews assigned to 5 airplanes. Thus, a chromosome of 3 genes could be used in this algorithm with each gene representing a crew on a certain plain.*

b) Suggest what could be the alphabet of this algorithm? What is its size?

Answer: *The alphabet of genes representing the crews can be used. Thus, its size is 5.*

c) Suggest a fitness function for this problem.

Answer: *You may come up with different versions, but it is important for the fitness to take into account the condition that cabin crews cannot work more than 2 days in a row. For example, the fitness function can take into account how many days each crew has left before a day off (e.g. 1 or 0). The fitness could be calculated as the sum of these numbers for all drivers in the chromosome.*

d) How many solutions are in this problem? Is it necessary to use Genetic Algorithms for solving it? What if the company operated more plains and employed more crews?

Answer: *The number of solutions is the number of times 3 crews can be selected out of 5 without replacement and without taking into account their order. The first crew can be selected in 5 different ways, the second in 4 ways and the third in 3 different ways. These numbers multiplied together will give us total number times how 3 crews can be selected randomly out of 5:* $5 \times 4 \times 3 = 60$ *times. However, there are 6 possible combinations in which 3 crews can be ordered, and because the order does not matter the answer is* $60/6 = 10$. *Thus, there are 10 possible solutions for this problem.*

It is not really necessary to use GA for a problem with such a small population, because solutions can be checked explicitly. However, as the number of crews and airplanes increases, so does the number of solutions, and the use of GA can be the only option. In fact, if n is the number of cabin crews and $k \leq n$ is the number of airplanes, then the number of solutions is

$$\frac{n!}{k!(n-k)!}.$$

For example, if the company operated 10 airplanes and employed 20 cabin crews, then the number of solutions would be

$$\frac{20!}{10!(20-10)!} = 184,756$$

What is the difference between crisp and fuzzy relations? Write the properties of fuzzy relations with example.

Crisp Set: Countability and finiteness are identical properties which are the collection objects of crisp set. '*X*' is a crisp set defined as the group of elements present over the universal set i.e. *U*. In this case a random element is present that may be a part of *X* or not that means two ways are possible to define the set. These are first element would become from set *X*, or it does not come from *X*.

Fuzzy Set: The Integration of the elements having a changing degree of membership in the set is called as fuzzy set. The word "*fuzzy*" indicates vagueness, On the other hand, we can say that the replacement among various degrees of the membership implies that the vague and ambiguity of the fuzzy set. Hence, the measurement of the membership of the elements from the universe in the set against a function for detecting the uncertainty and ambiguity.

S.No	Crisp Set	Fuzzy Set
1	Crisp set defines the value is either 0 or 1.	Fuzzy set defines the value between 0 and 1 including both 0 and 1.
2	It is also called a classical set.	It specifies the degree to which something is true.
3	It shows full membership	It shows partial membership.
4	Eg1. She is 18 years old. Eg2. Rahul is 1.6m tall	Eg1. She is about 18 years old. Eg2. Rahul is about 1.6m tall.
5	Crisp set application used for digital design.	Fuzzy set used in the fuzzy controller.
6	It is bi-valued function logic.	It is infinite valued function logic
7	Full membership means totally true/false, yes/no, 0/1.	Partial membership means true to false, yes to no, 0 to 1.

1. ADALINE + MADALINE
2. Backpropagation
3. Basic ANN & BOLAY.
4.
5.