Fuzzy Systems Assignments – Topic nr 3 – Fuzzy inference System

A fuzzy interface system is a system based on **fuzzy logic** that analyzes input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical logic, which operates on discrete values of either 1 or 0 (true or false, respectively). The term fuzzy refers to the fact that the logic involved can deal with concepts that cannot be expressed as the true or false but rather as partially true. Although alternative approaches such as genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to automatize tasks that are already successfully performed by humans.

Advantages of Fuzzy Logic Systems and why should we use them

- highly suitable for data with imprecision, distortion noise and nonlinearity.
- convenient way to map input space to the output space.
- flexible and easy to modify
- can use the knowledge and experience of experts
- can be combined with other control techniques
- the structure and math behind are relatively simple and understandable
- mimics the logic of human thoughts
- built on the structures of qualitative description used in everyday language, and therefor easy to use.
- As Lotfi Zadeh, who is considered to be the father of fuzzy logic, once remarked: "In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper."

Disadvantages of Fuzzy Logic Systems

- Many researchers proposed different ways to solve a given problem through fuzzy logic which lead to ambiguity. There is no systematic approach to neither system designing nor solving a given problem through fuzzy logic
- Sometimes accuracy of Fuzzy Logic Systems can be low, so they are suitable only for the problems which do not necessarily need that
- Validation and verification of a fuzzy knowledge-based system needs extensive testing
- Setting exact, fuzzy rules and, membership functions is a difficult task

Architecture of a Fuzzy Inference System

My Fuzzy Inference System consists of four main parts:

Rule Base:

It contains all the rules and the IF-THEN conditions provided by the experts to control the decision making system. Recent development in fuzzy theory offer several effective methods for the design and tuning of fuzzy controllers mainly used to reduce the number of fuzzy rules.

Fuzzification:

It is used to convert exact, crisp inputs passed into the control system for further processing into fuzzy sets.

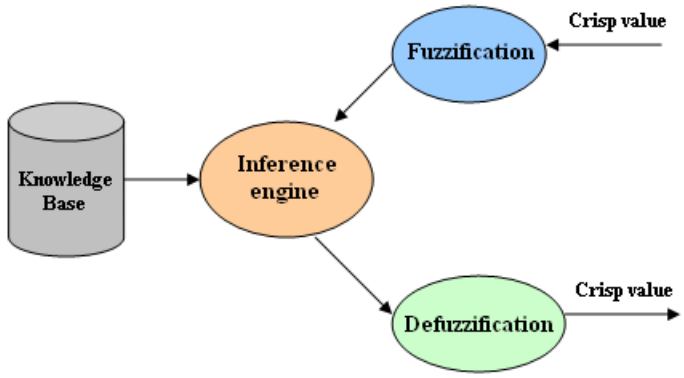
Inference Engine:

It determines the degree of match between the fuzzy input and each rule and also decides which

rules has to be fired according to the input field. Next, the fired rules are combined to form the control actions.

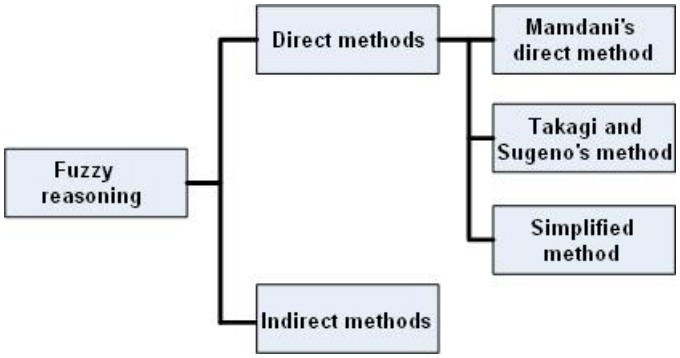
Defuzzification:

It is used to convert the fuzzy sets obtained by inference engine into a crisp value. There are several defuzzification methods available so the best suited one is used.



Classification of fuzzy inference methods

Fuzzy inference methods are classified into direct and indirect methods, where direct methods like Mamdani and Sugeno methods are the most commonly used. The main difference between Mamdani and Sugeno methods is that the Mamdani method output is a fuzzy set that needs defuzzification and Sugeno method output is either linear or constant.



Example use of Fuzzy Inference System using Mamdani method

An example use of a Mamdani inference system is shown below. To compute the output given the inputs, one must go through these steps:

Step 1: Define linguistic inputs and outputs

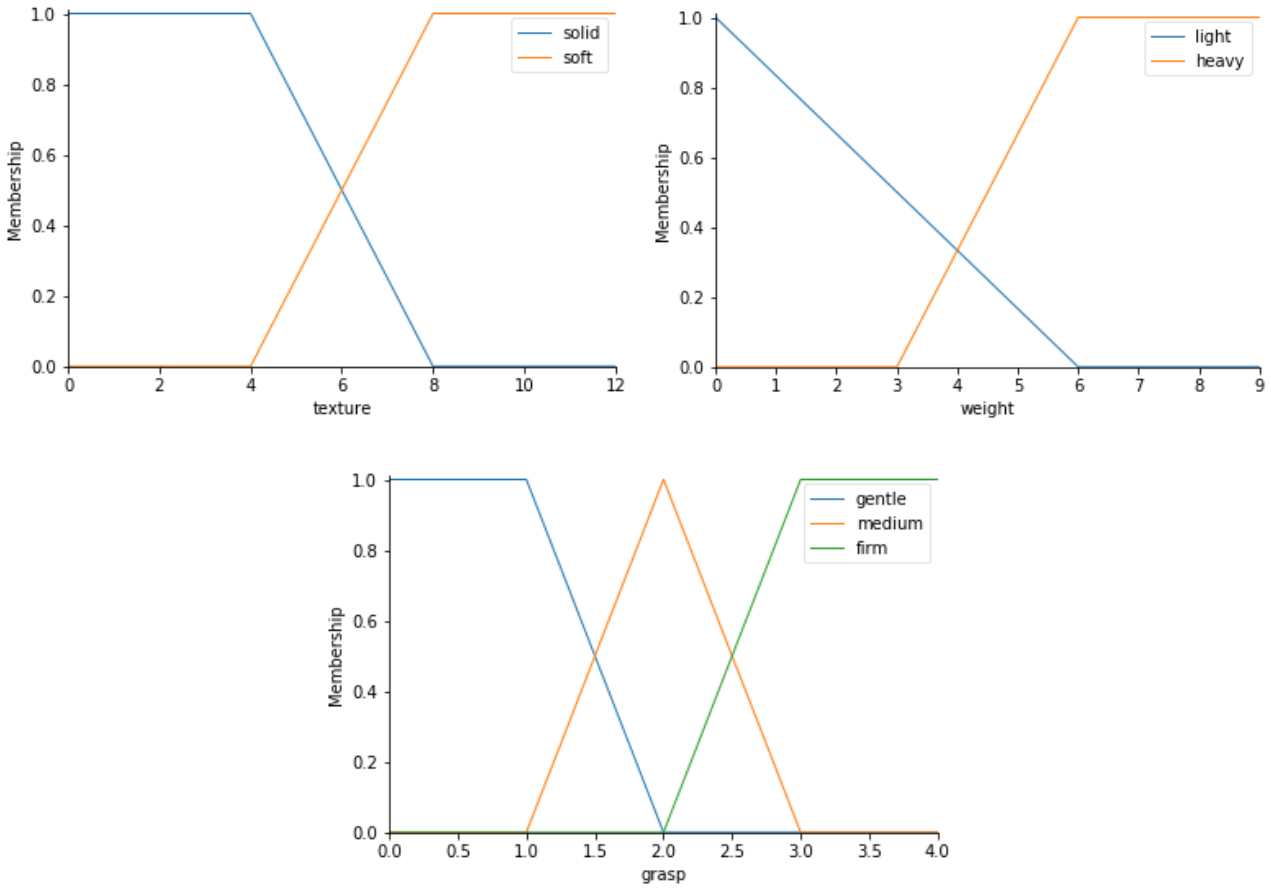
Linguistic variables are input and output variables in the form of simple words covering some part of overall values within some universe. Available values of my functions looks as follows:

TEXTURE(x) = { SOFT, SOLID }, WEIGHT(x) = { LIGHT, HEAVY },

GRIP = {GENTLE, MEDIUM, FIRM }

Step 2: Define membership functions

The most common shapes of membership functions are triangular and trapezoidal but the shape is generally less important than their number and placement.



There's more information about other membership functions in the next part of this report.

Step 3: Define knowledge base of rules

We going to define a collection of logic rules in the form of IF-THEN statements, where the IF part is called the antecedent and the THEN part is called the consequent. In practice rules usually have several antecedents that are combined using fuzzy operators, such as AND, OR, and NOT where AND most commonly uses the minimum weight of all the antecedents, OR uses the maximum value and there is also a NOT operator subtracts a membership function from 1 to give the complementary function.

RULE1: if texture is SOLID and weight is HEAVY then apply FIRM grasp

RULE2: if texture is SOLID and weight is LIGHT then apply MEDIUM grasp

RULE3: if texture is SOFT and weight is HEAVY then apply MEDIUM grasp

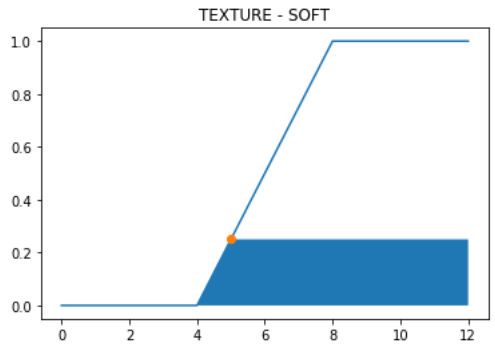
RULE4: if texture is SOFT and weight is LIGHT then apply GENTLE grasp

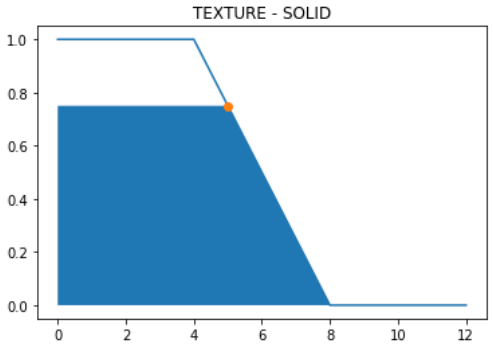
WEIGHT / *TEXTURE*	LIGHT	HEAVY
SOFT	GENTLE	MEDIUM
SOLID	MEDIUM	FIRM

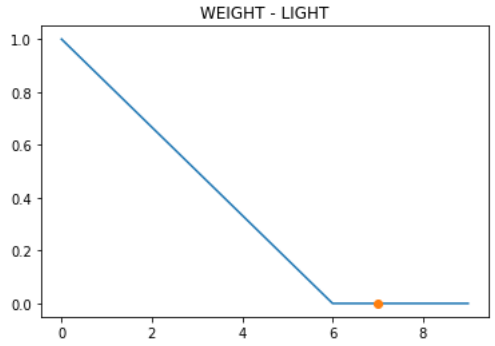
There's more information about other fuzzy operators in the next part of this report.

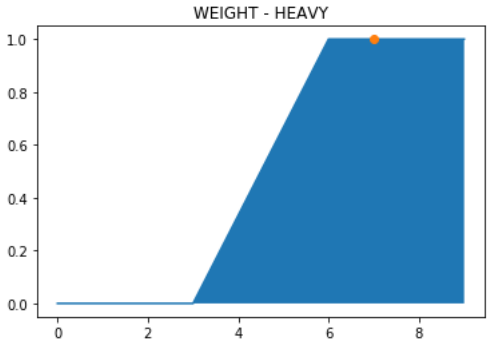
Step 4: Convert crisp data into fuzzy data sets

In this step our system maps crisp input values to the appropriate membership functions and truth values. The process of converting a crisp value to a fuzzy value is called **fuzzification.**





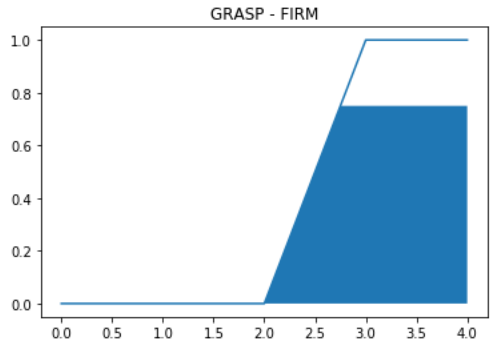




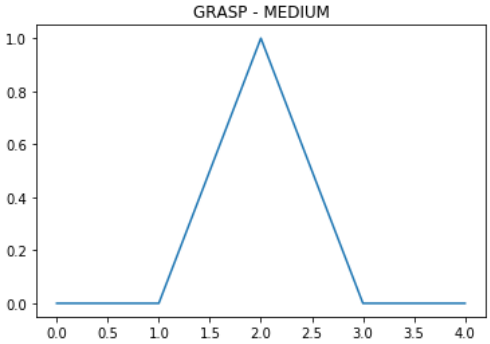
Step 5: Evaluate rules in the rule base

Given mappings of input variables into membership functions and truth values our system invokes rules and generates membership functions and truth values for all the rules.

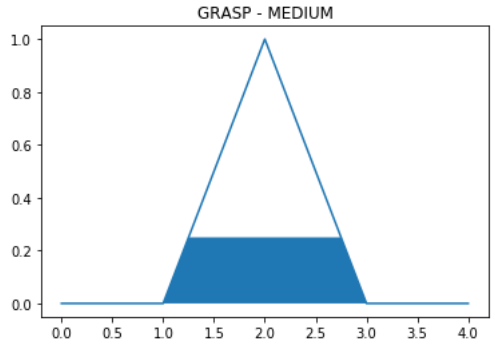
RULE1



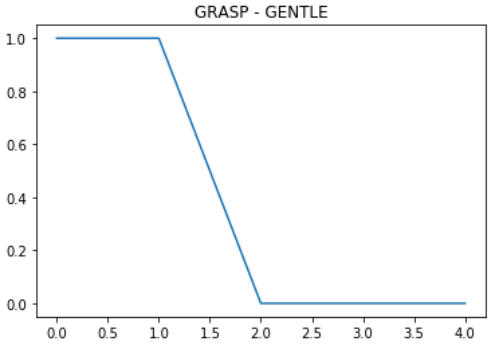
RULE2



RULE3



RULE4

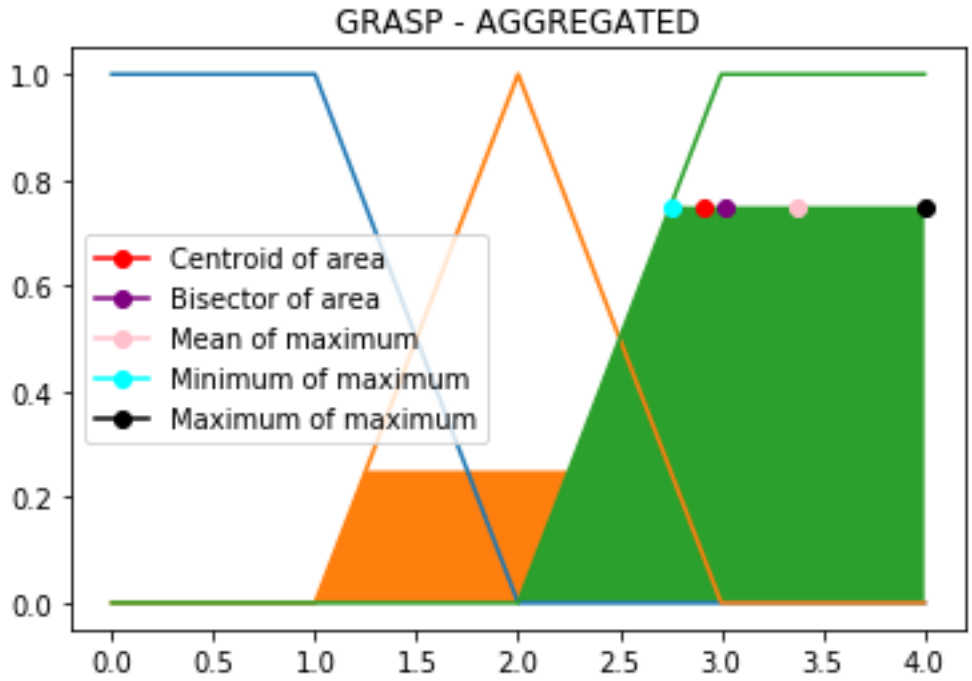


Step 6: Combine results from each rule and convert output data into non-fuzzy value

Membership functions and truth values of rules are combined into a membership function controlling the output variable. Then our system converts it to a crisp value in a procedure known as defuzzification that can be achieved using several different methods:

Centroid method – choosing crisp value using the center of mass.
Bisector method – choosing crisp value so that area limited by the membership function on the left side would be equal to the area limited by the membership function on the right side.
Mean, minimum and maximum method – choosing crisp value using the mean, minimum and maximum value of the highest rule.

The centroid method favors the rules with greatest area, while the mean, minimum and maximum methods favors the rules with the greatest output value.



Centroid	Bisector	Mean	Minimum	Maximum
2.91	3.02	3.375	2.75	4.0

Summary and conclusions

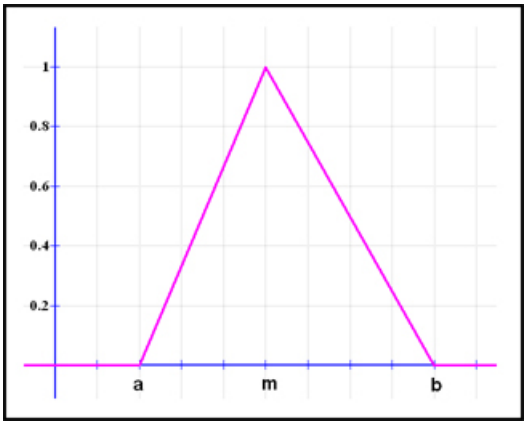
It is worth mentioning that Mamdani method is useful when there is a small number of variables. Otherwise we could encaunter some of the following difficulties:
-The number of rules increases exponentially with the number of variables in the antecedent,
-The more rules we construct, the harder is to know if they are suitable for our problem,
-If the number of variables in the antecedent is too large, it will be difficult to grasp the casual relationship between the antecedents and the consequents and therefore constructing new rules will become harder

Membership functions

Membership functions allow you to quantify linguistic term and represent a fuzzy set. A membership function for a fuzzy set A on the universe of discourse X is defined as $\mu_A(X)$: $X \rightarrow [0,1]$. Here, each element of X is mapped to a value between 0 and 1. It is called membership value or degree of membership. It quantifies the degree of membership of the element in X to the fuzzy set
x axis represents the universe of discourse.
y axis represents the degrees of membership in the [0, 1] interval. There can be multiple membership functions applicable to fuzzify a numerical value.
The most popular membership functions are:

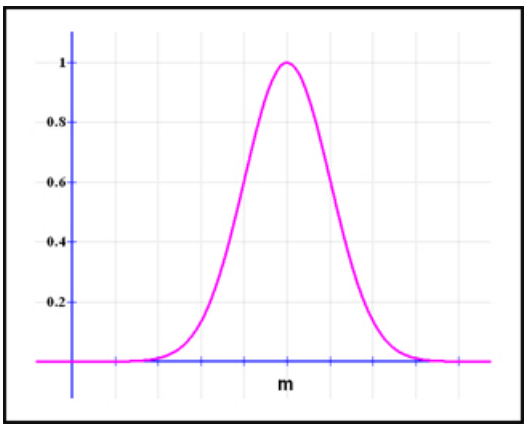
Triangular function

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{m-a}, & a < x \le m \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x \ge b \end{cases}$$



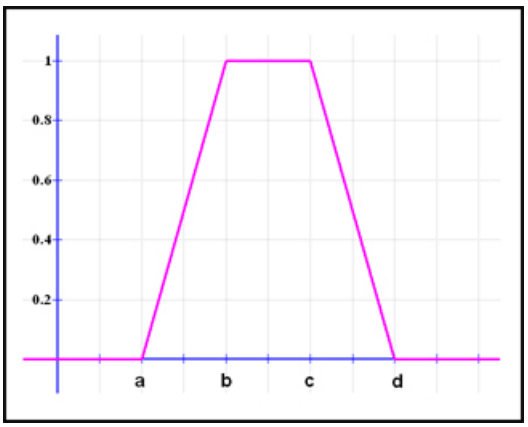
Gaussian function

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}}$$



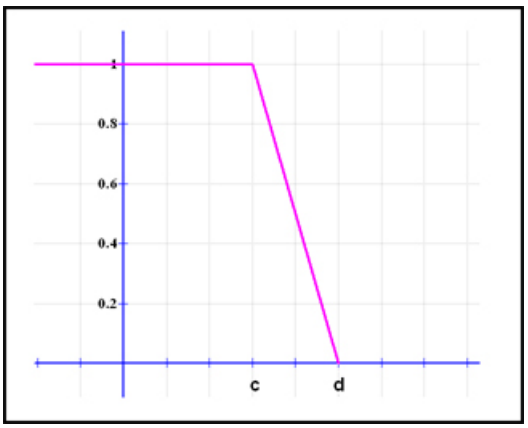
Trapezoidal function

$$\mu_A(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases}$$



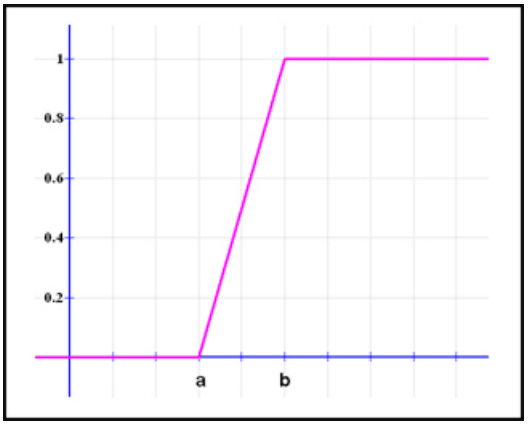
R-function

$$\mu_A(x) = \begin{cases} 0, & x > d \\ \frac{d-x}{d-c}, & c \le x \le d \\ 1, & x < c \end{cases}$$



L-function

$$\mu_A(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases}$$



Singleton function

FUZZY OPERATORS

T-norms are binary operations that generalize intersection of two fuzzy sets such as
T: [0,1] x [0,1] → [0,1] which satisfies the following properties:
Commutativity: T(a, b) = T(b, a)
Associativity: T(a, T(b, c)) = T(T(a, b), c)
Identity element: T(a,1) = T(1,a) = a
Monotonicity: if a ≤ c and b ≤ d then T(a, b) ≤ T(c, d)

Some examples of t-norms are:

Minimum t-norm: T(a, b) = min(a, b)

Algebraic product: T(a, b) = a*b

Bounded difference: T(a, b) = max(0, a+b-1)

Einstein t-norm: T(a, b) = (a*b)/(2-(a+b-a*b))

Hamacher t-norm: T(a, b) = (a*b)/(a+b-a*b)

Drastic t-norm: $T(a,b)=\begin{matrix} min(a,b), & if\ max(a,b)=1 \\ 0, & if\ max(a,b)\neq 1 \end{matrix}$

Nilpotent t-norm: $T(a,b)=\begin{matrix} min(a,b), & if\ a+b>1 \\ 0, & if\ a+b\leq 1 \end{matrix}$

T-conorm (also called S-norms) are binary operations that generalize union of two fuzzy sets such as
S: [0,1] x [0,1] → [0,1] which satisfies the following properties:
S(a, b) = 1 – T(1 – a, 1 – b)
Commutativity: S(a, b) = S(b, a)
Associativity: S(a, S(b, c)) = S(S(a, b), c)
Identity element: S(a,0) = S(0,a) = a
Monotonicity: if a ≤ c and b ≤ d then S(a, b) ≤ S(c, d)

Some examples of t-conorms are:

Maximum t-conorm: S(a, b) = max(a, b)

Probabilistic sum: S(a, b) = a+b-a*b

Bounded sum: S(a, b) = min(1, a+b)

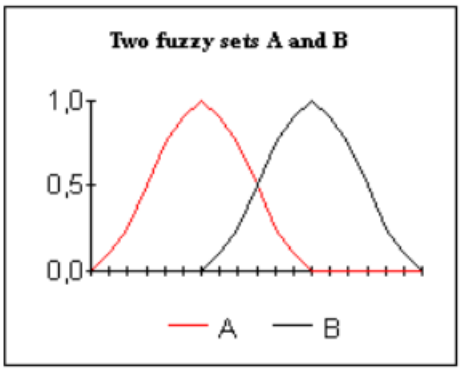
Einstein t-conorm: S(a, b) = (a+b)/(1+a*b)

Hamacher t-conorm: S(a, b) = (a+b-2*a*b)/(1-a*b)

Drastic t-conorm: $S(a,b)=\begin{matrix} max(a,b), & if\ min(a,b)=0 \\ 1, & if\ min(a,b)\neq 0 \end{matrix}$

Nilpotent t-norm: $S(a,b)=\begin{matrix} max(a,b), & if\ a+b<1 \\ 1, & if\ a+b\geq 1 \end{matrix}$

Below, on another page, there's a visualization of some t-norms and t-conorms mentioned above:



The red lines are the T-norms and the blue lines are the S-norms.