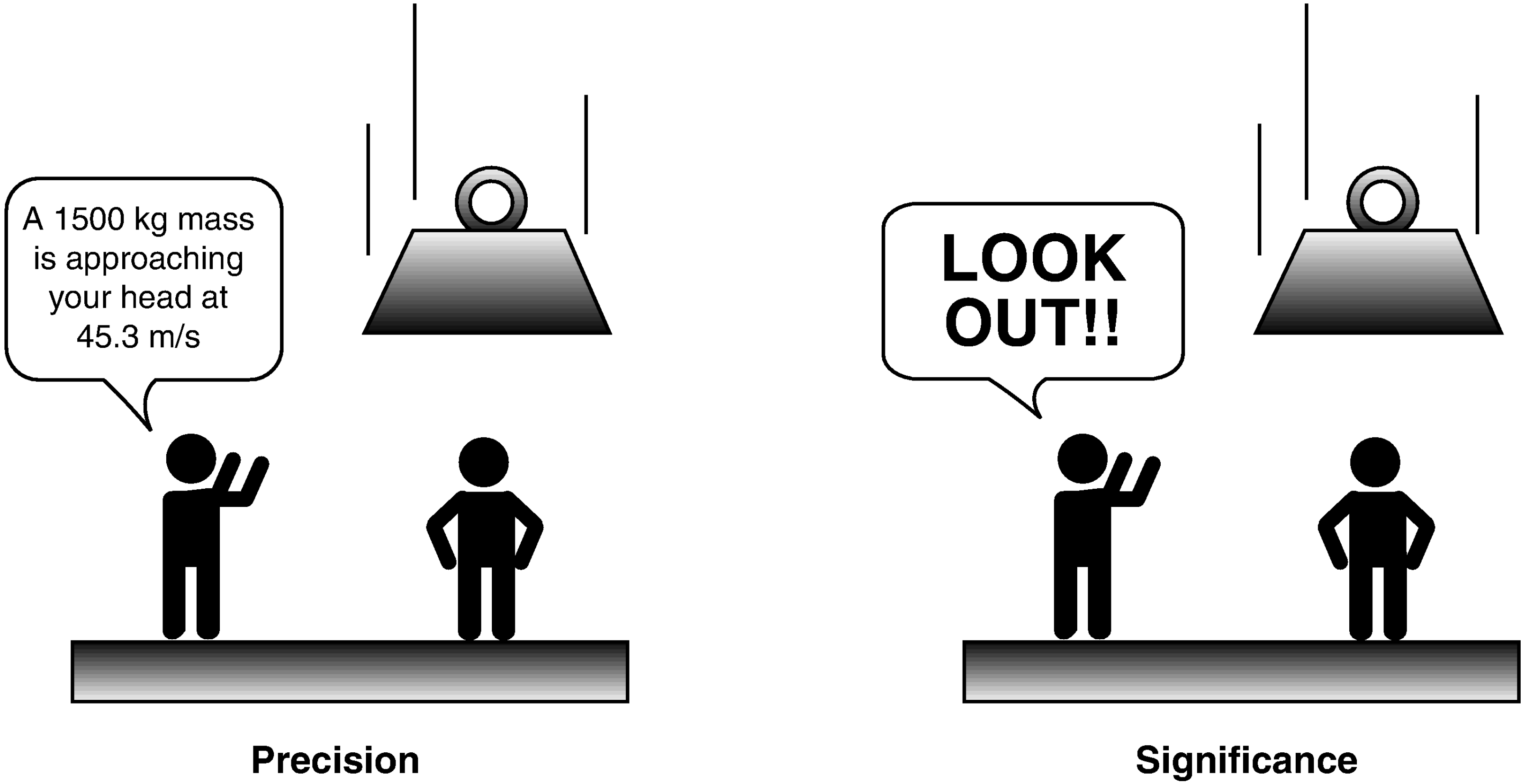
11 Foundations of Fuzzy Logic

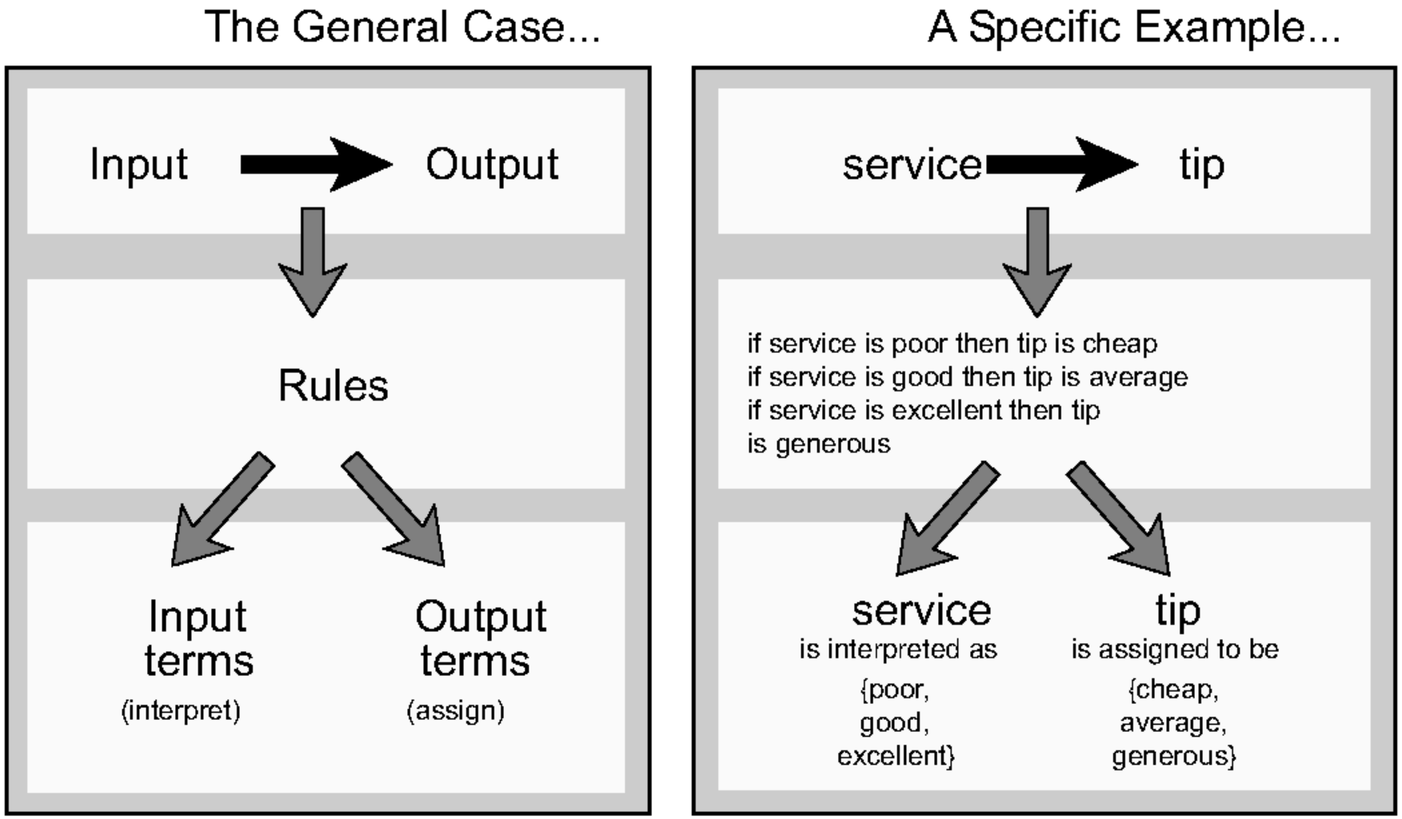
(based on *Fuzzy Logic Toolbox. User's Guide.*)

- Processing of vague and imprecise data is our everyday experience
- Consider the problem of Precision and Significance in the real world:



- Fuzzy logic is a convenient way to map an input space to an output space.

- Fuzzy logic is about **mapping an input space to an output space**, and the primary mechanism for doing this is a list of if-then statements called rules.
- All rules are evaluated in parallel, and the order of the rules is unimportant.
- The rules refer to variables and the adjectives that describe those variables.



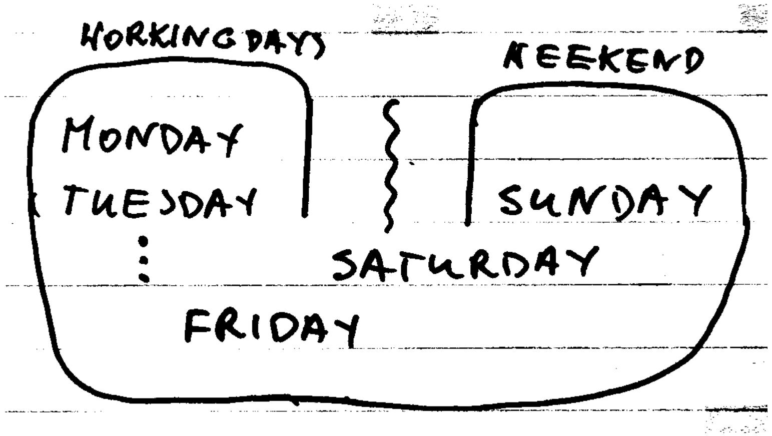
Fuzzy inference is a method that interprets the values in the input vector and,

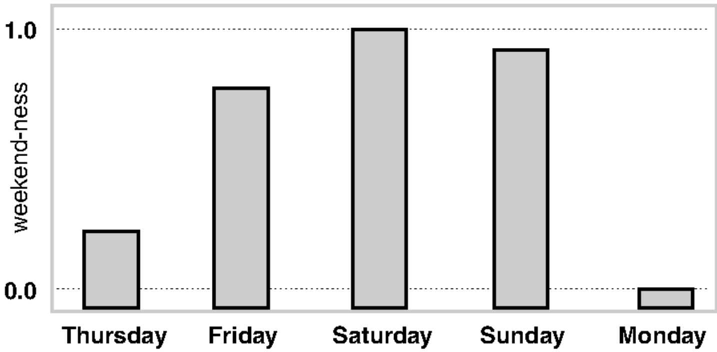
based on some set of rules,

assigns values to the output vector.

Fuzzy sets

- A fuzzy set, as opposed to the "classical set" does not have crisp, clearly defined boundary.
- Partitioning the set of days of the week into working days and weekend we might find problems with allocating Friday and Saturday crisply to only one subset.



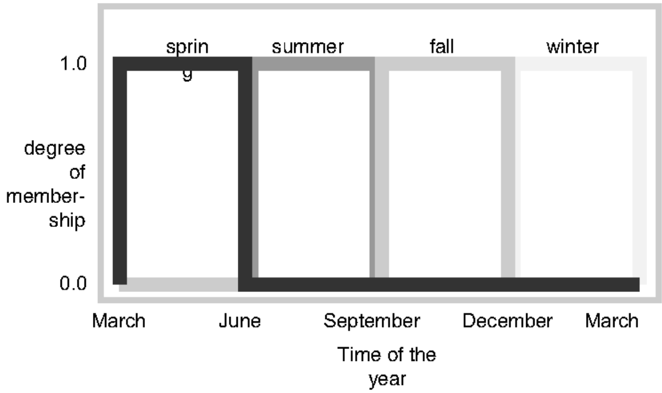


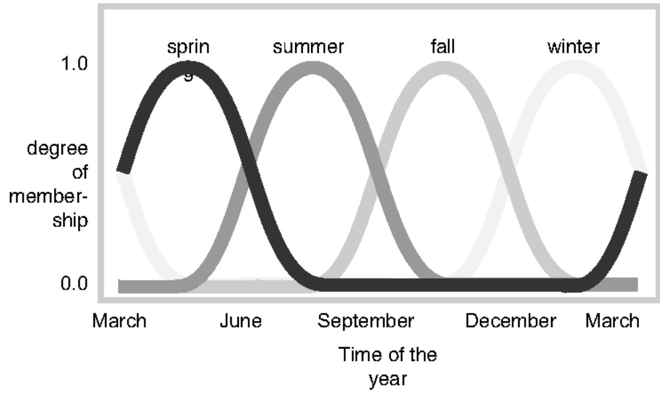
Days of the weekend multivalued membership

- In fuzzy logic, the truth of any statement becomes a matter of degree.
- Reasoning in fuzzy logic we give assign a degree of true to each statement, for example
- a degree of weekend-ness for every day of the week.
- The plot shows how much a particular day can be classified as a weekend
- The function that defines the weekend-ness of any instant in time maps the input space (time of the week) to the output space (weekend-ness).
- It is known as a **membership function**.

Membership Functions

- A **membership function** (MF, or μ) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1.
- The input space is sometimes referred to as the universe of discourse.
- Consider the membership functions for seasons of the year

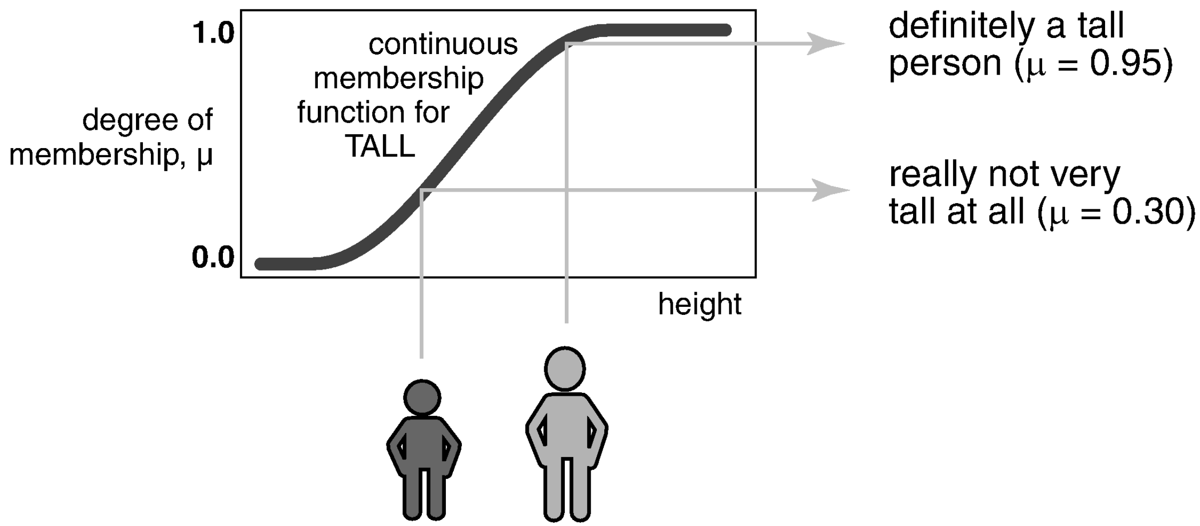




- Using the astronomical definitions for the season, we get sharp boundaries as shown on the left in the figure.
- But what we experience as the seasons vary more or less continuously as shown on the right plot(in temperate northern hemisphere climates).

Membership Functions

- A classical set is defined by a crisp membership for example
$$A = \{x | x > 6\}$$
- A fuzzy set is an extension of a classical set where the membership function describes a degree of belonging.
- If X is the universe of discourse and its elements are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs.
$$A = \{x, \mu_A(x) | x \in X\}$$
$\mu_A(x)$ is called the membership function of x in A.
- The membership function maps each element of X to a membership value between 0 and 1.



Membership Functions

- The Fuzzy Logic Toolbox includes 11 built-in membership function types

  built from several basic functions:

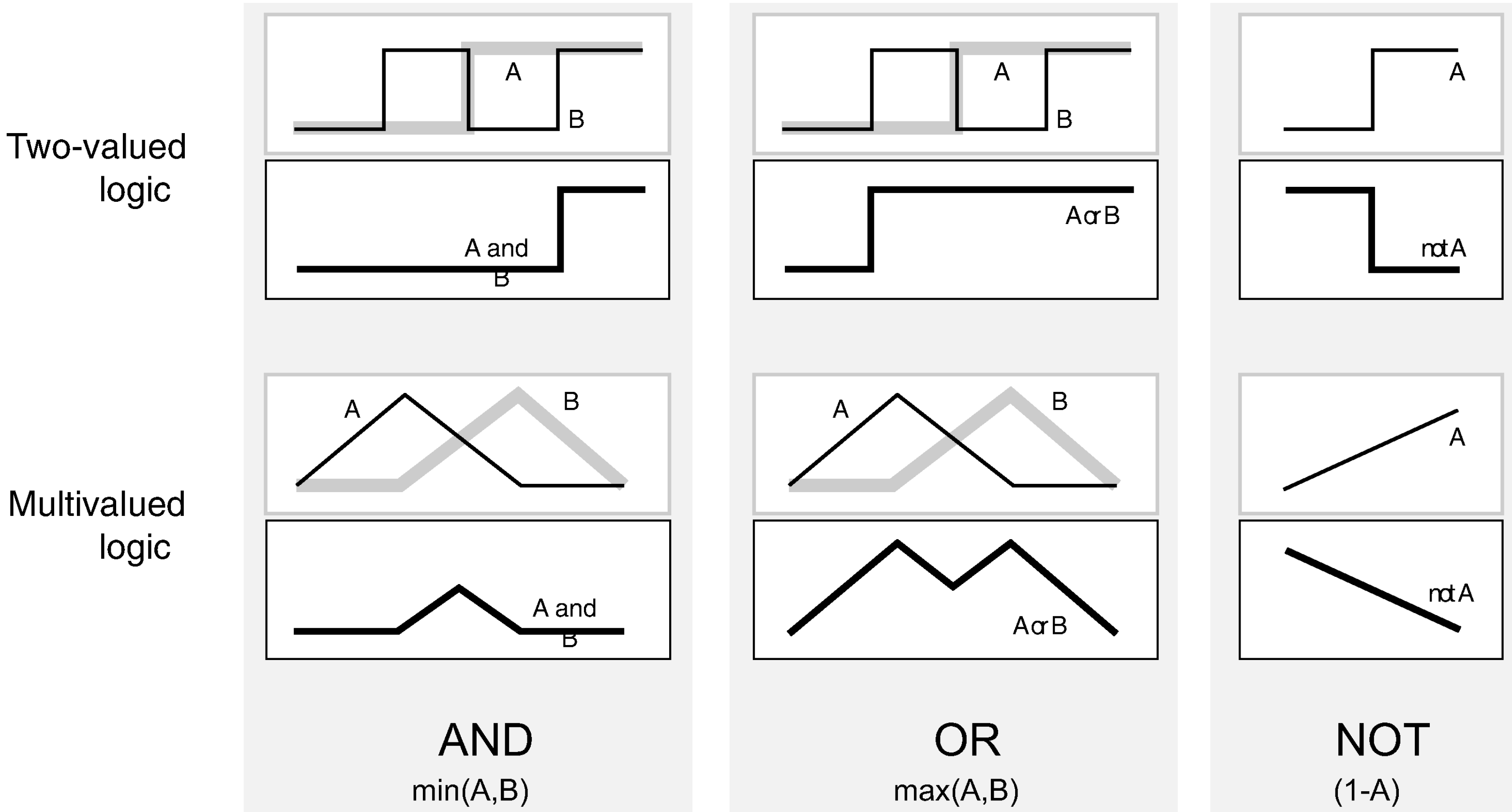
  piecewise linear functions, the Gaussian distribution function, the sigmoid curve, and quadratic and cubic polynomial curves.

Summary:

- Fuzzy sets describe vague concepts (fast runner, hot weather, weekend days).
- A fuzzy set admits the possibility of partial membership in it. (Friday is sort of a weekend day, the weather is rather hot).
- The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1. (Friday is a weekend day to the degree 0.8).
- A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.

Logic Operations

- Fuzzy logical reasoning is a superset of standard Boolean logic.
- If we keep the fuzzy values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold.



- Given these three functions, we can resolve any construction using fuzzy sets and the fuzzy logical operation AND, OR, and NOT.

If-Then rules

- Fuzzy sets and fuzzy operators are the **subjects** and **verbs** of fuzzy logic.
- The if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.
- A single fuzzy if-then rule assumes the form

  if x **is** A **then** y **is** B

  where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively.
- The **if**-part of the rule x is A is called the antecedent or premise, while the **then**-part of the rule y is B is called the consequent or conclusion.
- An example of such a rule might be

  if service == good **then** tip = average
- The input to an if-then rule is the current value for the input variable (in this case, service) and the output is an entire fuzzy set (in this case, average).
- This set will later be defuzzified, assigning one value to the output.

Interpreting an if-then rule

involves distinct parts:

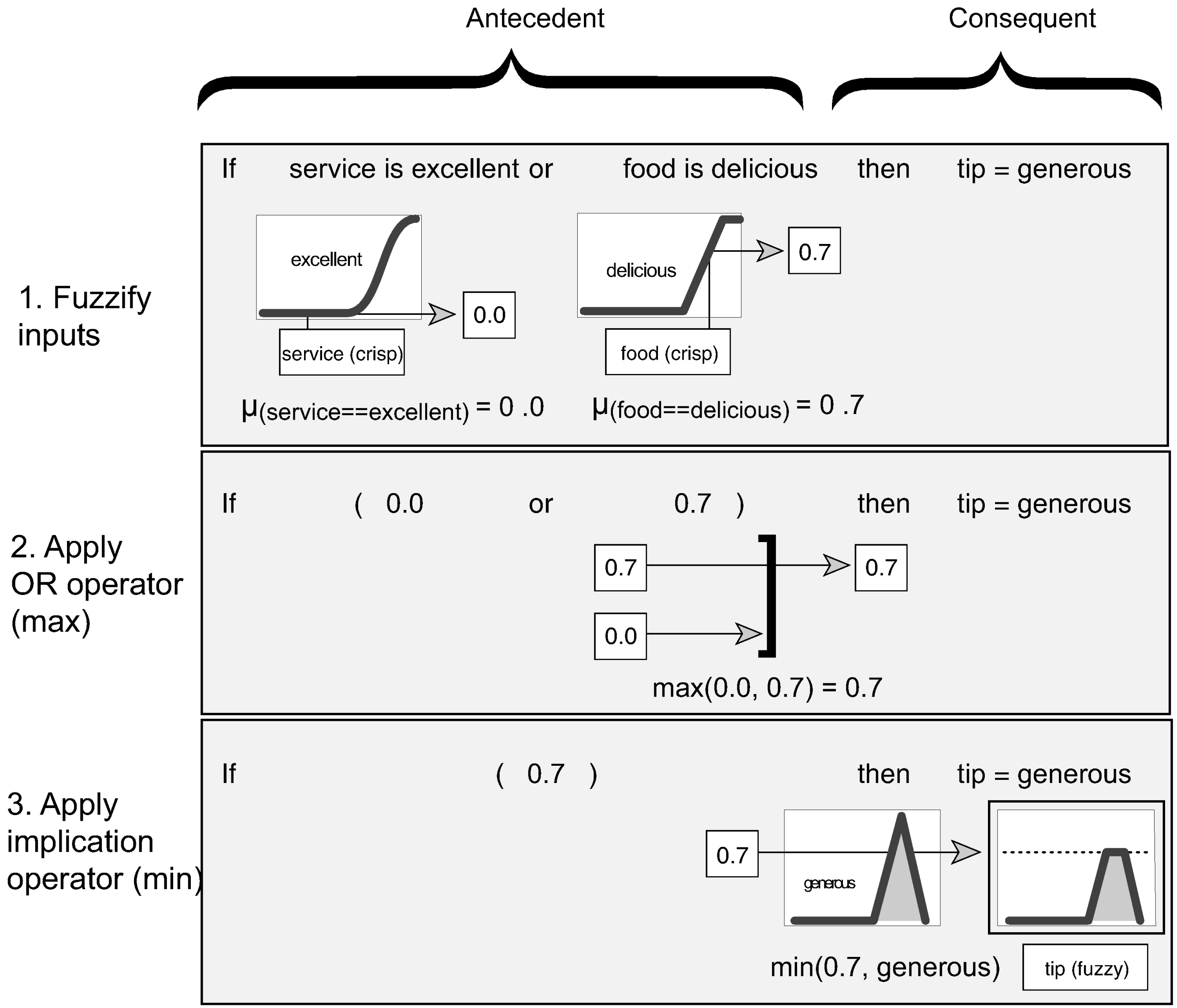
- first evaluating the antecedent (which involves fuzzifying the input and applying any necessary fuzzy operators) and
- second applying that result to the consequent (known as implication).
- If the antecedent is a fuzzy statement true to some degree of membership, then the consequent is also true to that same degree.
- The antecedent of a rule can have multiple parts:

  if sky is gray and wind is strong and barometer is falling, then ...

  in which case all parts of the antecedent are calculated simultaneously and resolved to a single number using the logical operators described in the preceding section.
- The consequent of a rule can also have multiple parts.

  if temperature is cold then hot water valve is open and cold water valve is shut

  in which case all consequents are affected equally by the result of the antecedent.



- The consequent specifies a fuzzy set be assigned to the output.
- The implication function then modifies that fuzzy set to the degree specified by the antecedent.
- The most common ways to modify the output fuzzy set are truncation using the **min** function

  (where the fuzzy set is chopped off as shown)

Interpreting if-then rules is a three-part process:

1. Fuzzify inputs: Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1.

   If there is only one part to the antecedent, this is the degree of support for the rule.

2. Apply fuzzy operator to multiple part antecedents:

   If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule.

3. Apply implication method:

   Use the degree of support for the entire rule to shape the output fuzzy set.

   The consequent of a fuzzy rule assigns an entire fuzzy set to the output.

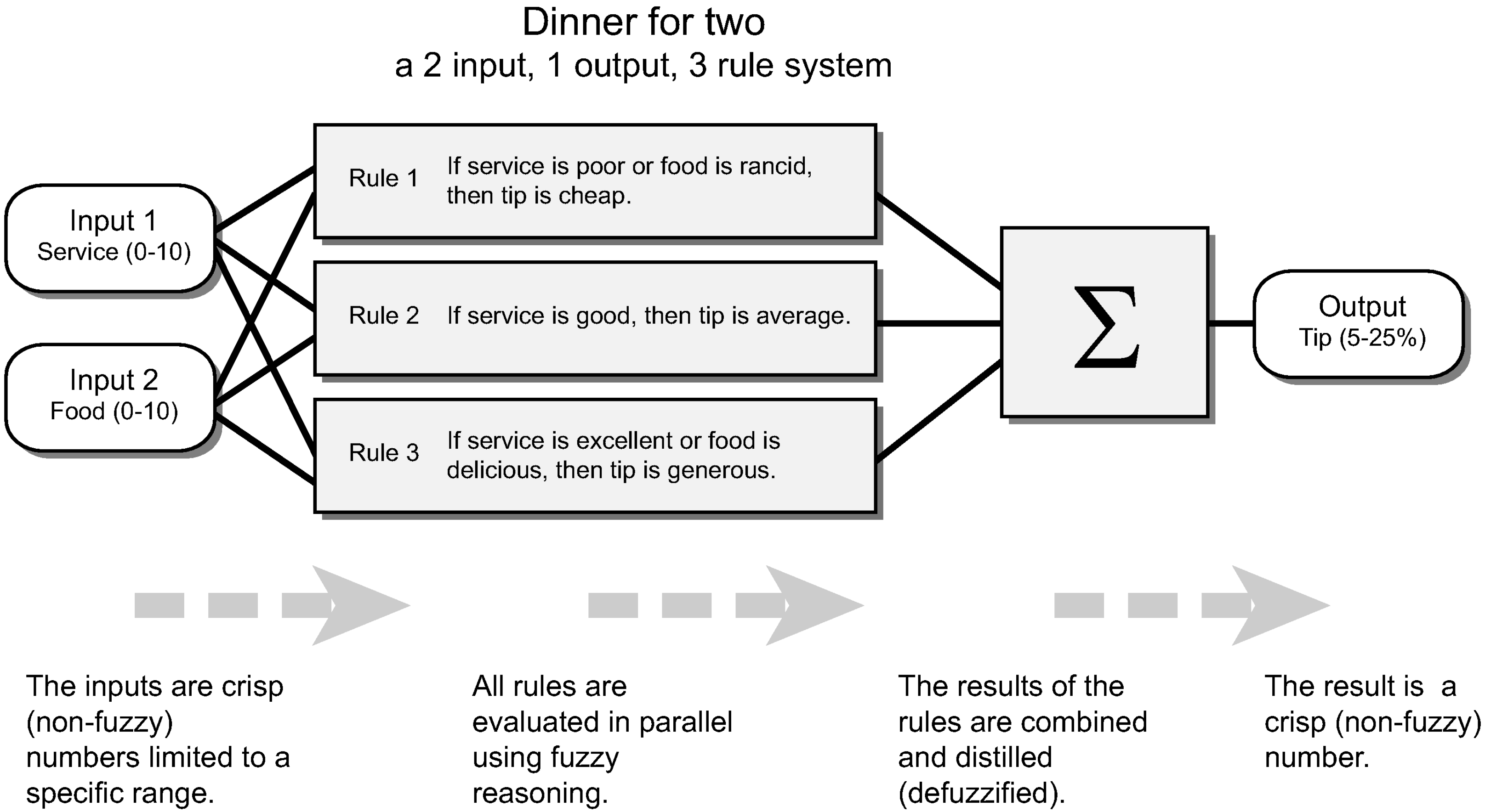
   This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent.

   If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

In general, one rule by itself doesnt do much good. Whats needed are two or more rules that can play off one another. The output of each rule is a fuzzy set.
The output fuzzy sets for each rule are then aggregated into a single output fuzzy set.
Finally the resulting set is defuzzified, or resolved to a single number.

Example: Dinner for Two

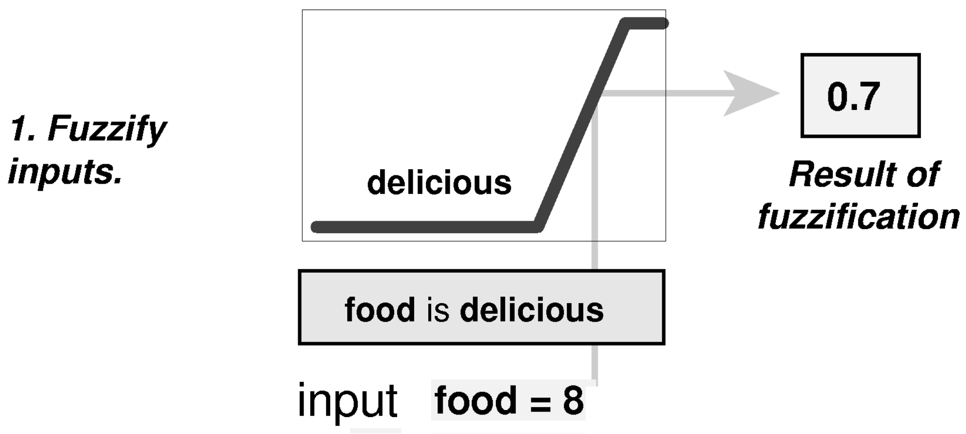
Consider a two-input, one-output, three-rule tipping problem as shown in the diagram below.



- The parallel nature of the rules is one of the more important aspects of fuzzy logic systems.
- Instead of sharp switching between modes based on breakpoints, we will glide smoothly from regions where the systems behavior is dominated by either one rule or another.

Step 1. Fuzzify Inputs

- The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions.
- Fuzzification of the input amounts to either a table lookup or a function evaluation.
- The example were using in this section is built on three rules, and each of the rules depends on resolving the inputs into a number of different fuzzy linguistic sets: service is poor, service is good, food is rancid, food is delicious, and so on.
- Before the rules can be evaluated, the inputs must be fuzzified according to each of these linguistic sets.



- Fuzzification of the food quality.
- It shows how well the food (rated on a scale of 0 to 10) qualifies, (via its membership function), as the linguistic variable delicious.
- We rated the food as an 8, which, given our graphical definition of delicious, corresponds to $\mu = 0.7$ for the delicious membership function.

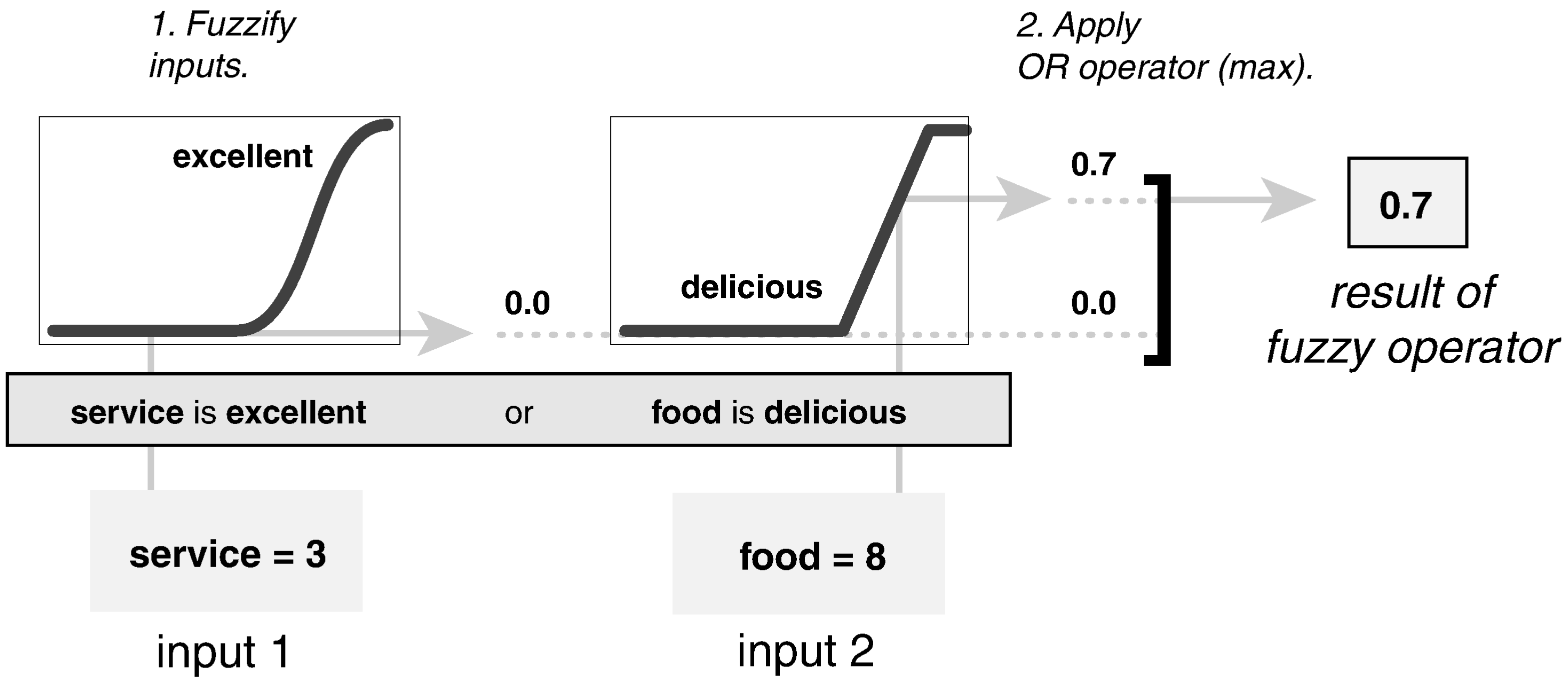
- In this manner, each input is fuzzified over all the qualifying membership functions required by the rules.

Step 2. Apply Fuzzy Operator

- Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule.
- If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule.
- This number will then be applied to the output function.
- The input to the fuzzy operator is two or more membership values from fuzzified input variables.
- The output is a single truth value.
- In the Fuzzy Logic Toolbox, two built-in AND methods are supported: min (minimum) and prod (product).
- Two built-in OR methods are also supported: max (maximum), and the probabilistic OR method **probor**.
- The probabilistic OR method (also known as the algebraic sum) is calculated according to the equation

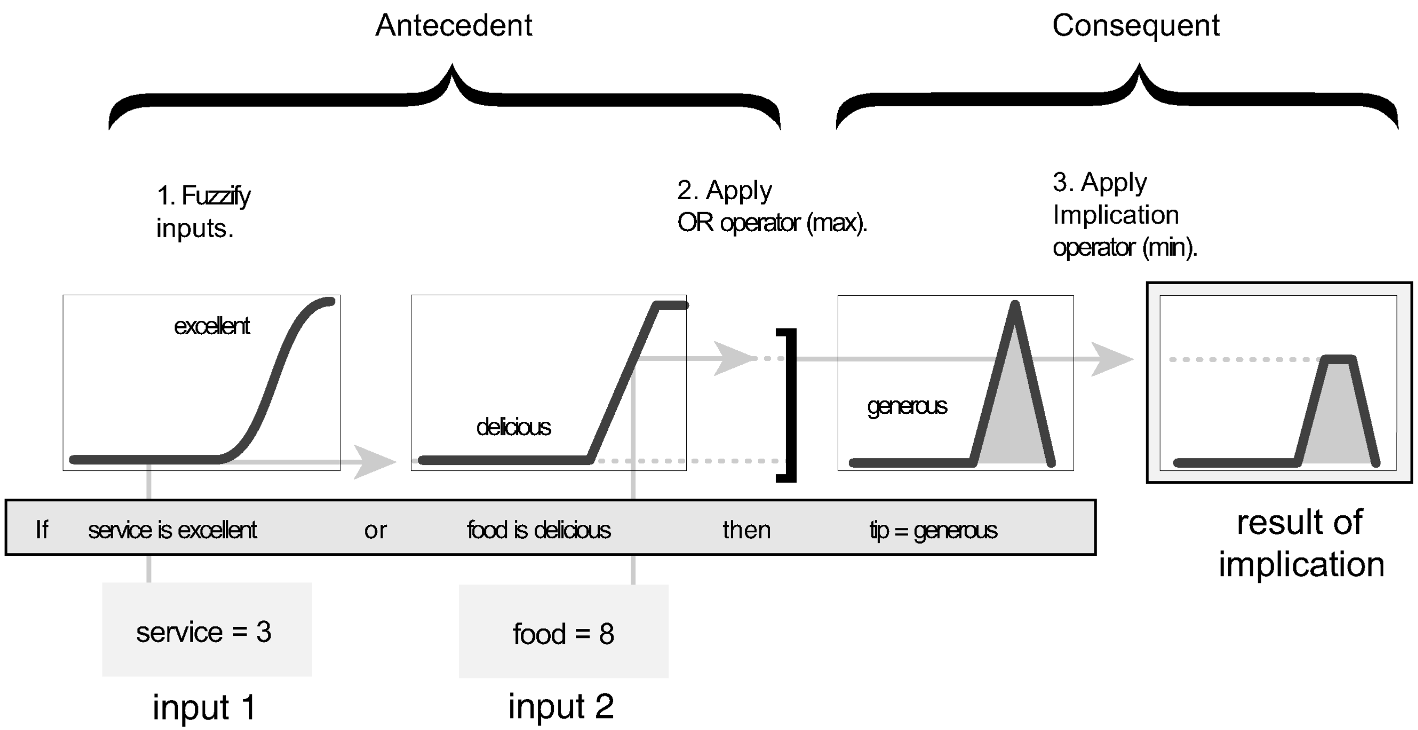
$$\text{probor}(a, b) = a + b - ab$$

An example of the OR operator **max** at work.



- Were evaluating the antecedent of the rule 3 for the tipping calculation.
- The two different pieces of the antecedent (service is excellent and food is delicious) yielded the fuzzy membership values 0.0 and 0.7 respectively.
- The fuzzy OR operator simply selects the maximum of the two values, 0.7, and the fuzzy operation for rule 3 is complete.
- If we were using the probabilistic OR method, the result would still be 0.7 in this case.

Step 3. Apply Implication Method

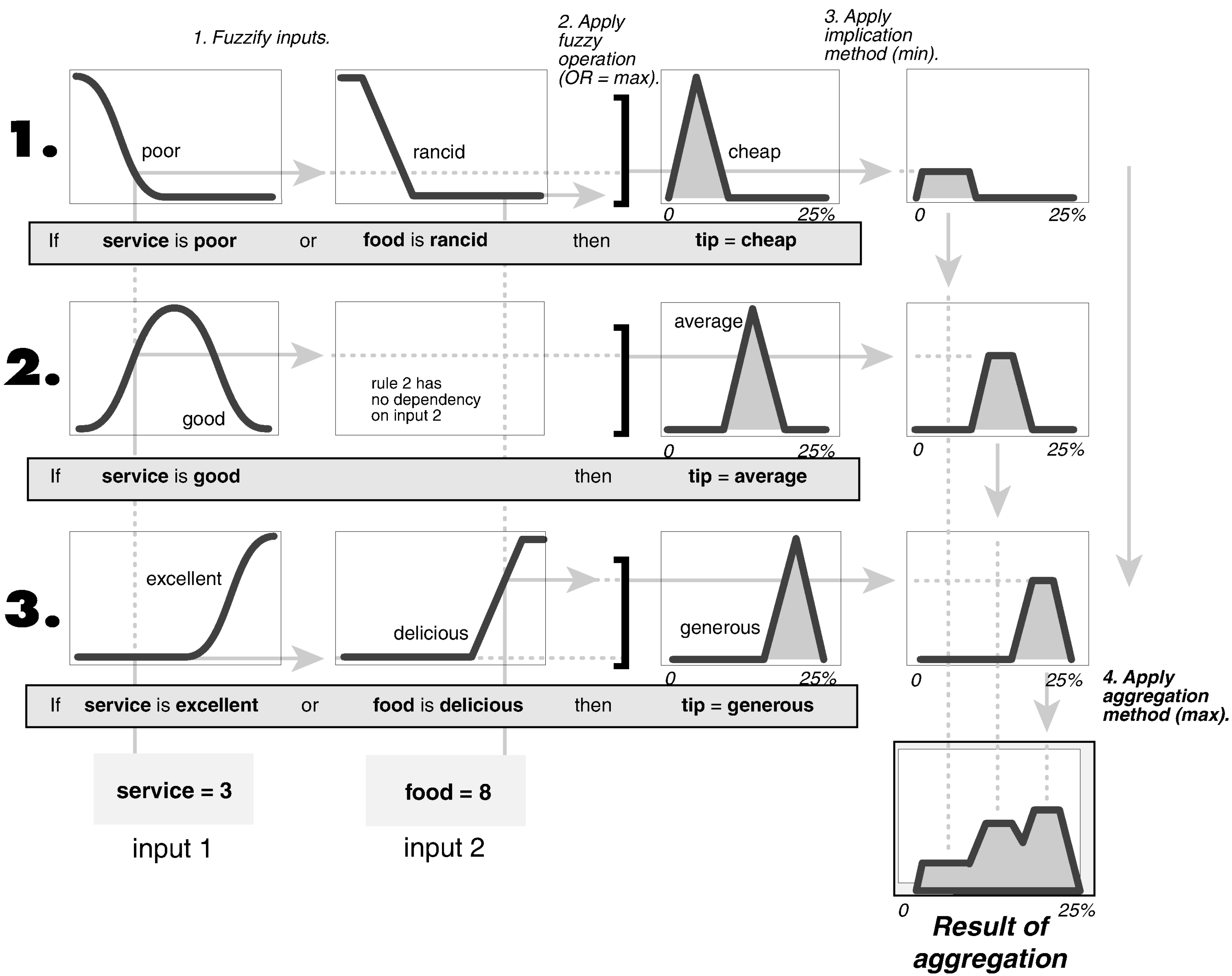


- A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it.
- The consequent is reshaped using a function associated with the antecedent (a single number).

- The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.
- Two built-in methods are supported, and they are the same functions that are used by the AND method: min (minimum), which truncates the output fuzzy set, and prod (product), which scales the output fuzzy set.

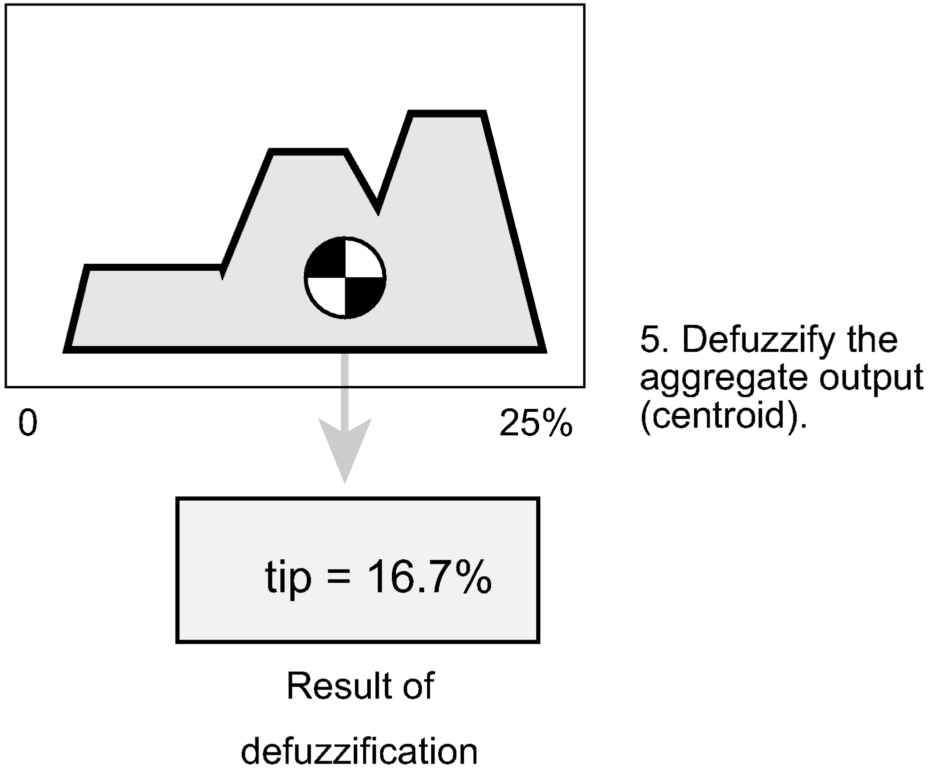
Step 4. Aggregate All Outputs

- Since decisions are based on the testing of all of the rules in an FIS, the rules must be combined in some manner in order to make a decision.
- Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set.
- Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification.
- The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule.
- The output of the aggregation process is one fuzzy set for each output variable.
- Notice that as long as the aggregation method is commutative (which it always should be), then the order in which the rules are executed is unimportant.
- Three built-in methods are supported: max (maximum), probor (probabilistic OR), and sum (simply the sum of each rules output set).
- In the following diagram, all three rules have been placed together to show how the output of each rule is combined, or aggregated, into a single fuzzy set whose membership function assigns a weighting for every output (tip) value.


1. Fuzzify inputs.
2. Apply fuzzy operation (OR = max).
3. Apply implication method (min).
1.
poor
rancid
cheap
0
25%
0
25%
If service is poor or food is rancid then tip = cheap
2.
good
rule 2 has no dependency on input 2
average
0
25%
0
25%
If service is good then tip = average
3.
excellent
delicious
generous
0
25%
0
25%
4. Apply aggregation method (max).
If service is excellent or food is delicious then tip = generous
service = 3
input 1
food = 8
input 2
0
25%
Result of aggregation

Step 5. Defuzzify

- The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number.
- As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number.
- However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

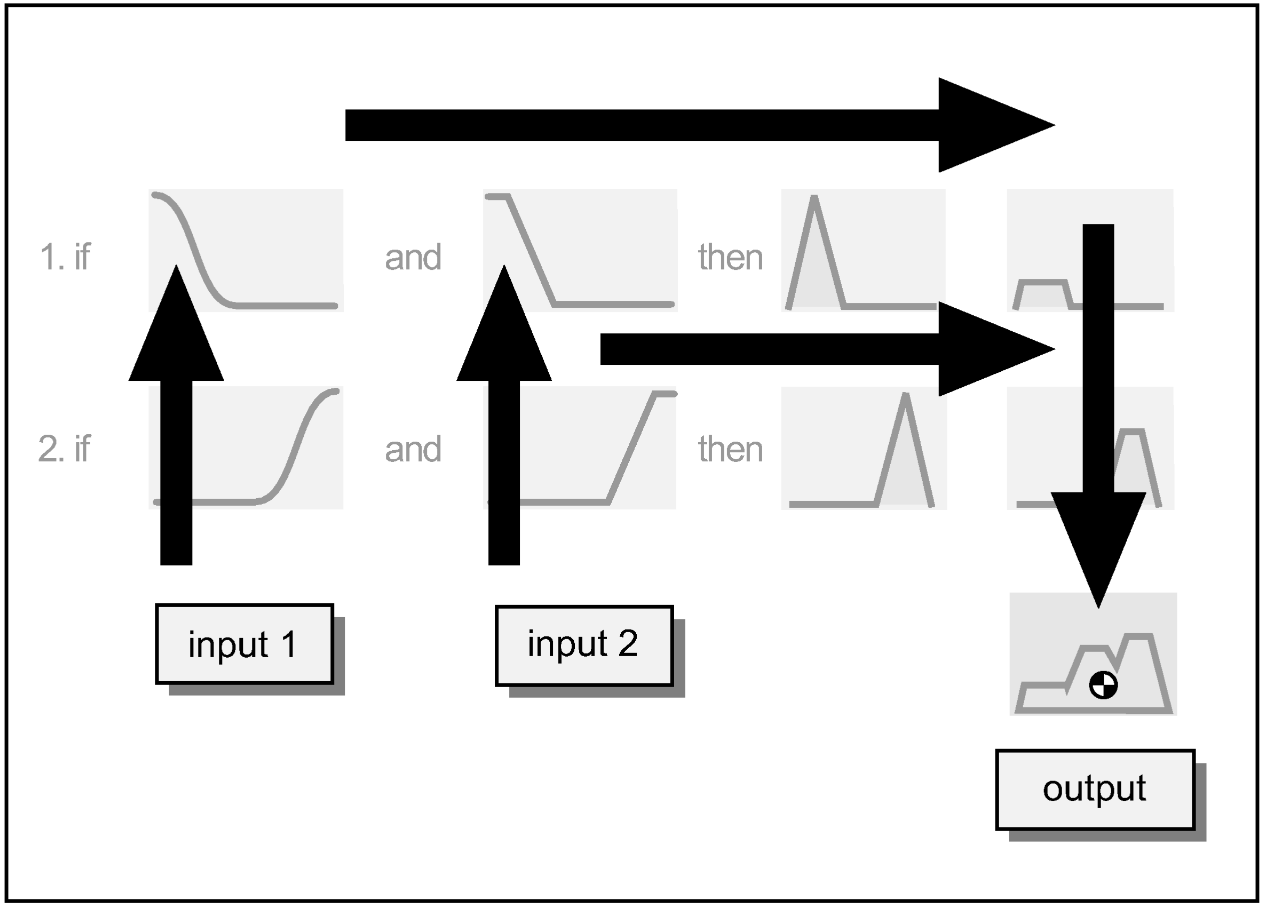


- The most popular defuzzification method is the centroid calculation, which returns the center of area under the curve.
- There are five built-in methods supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum.

The Fuzzy Inference Diagram

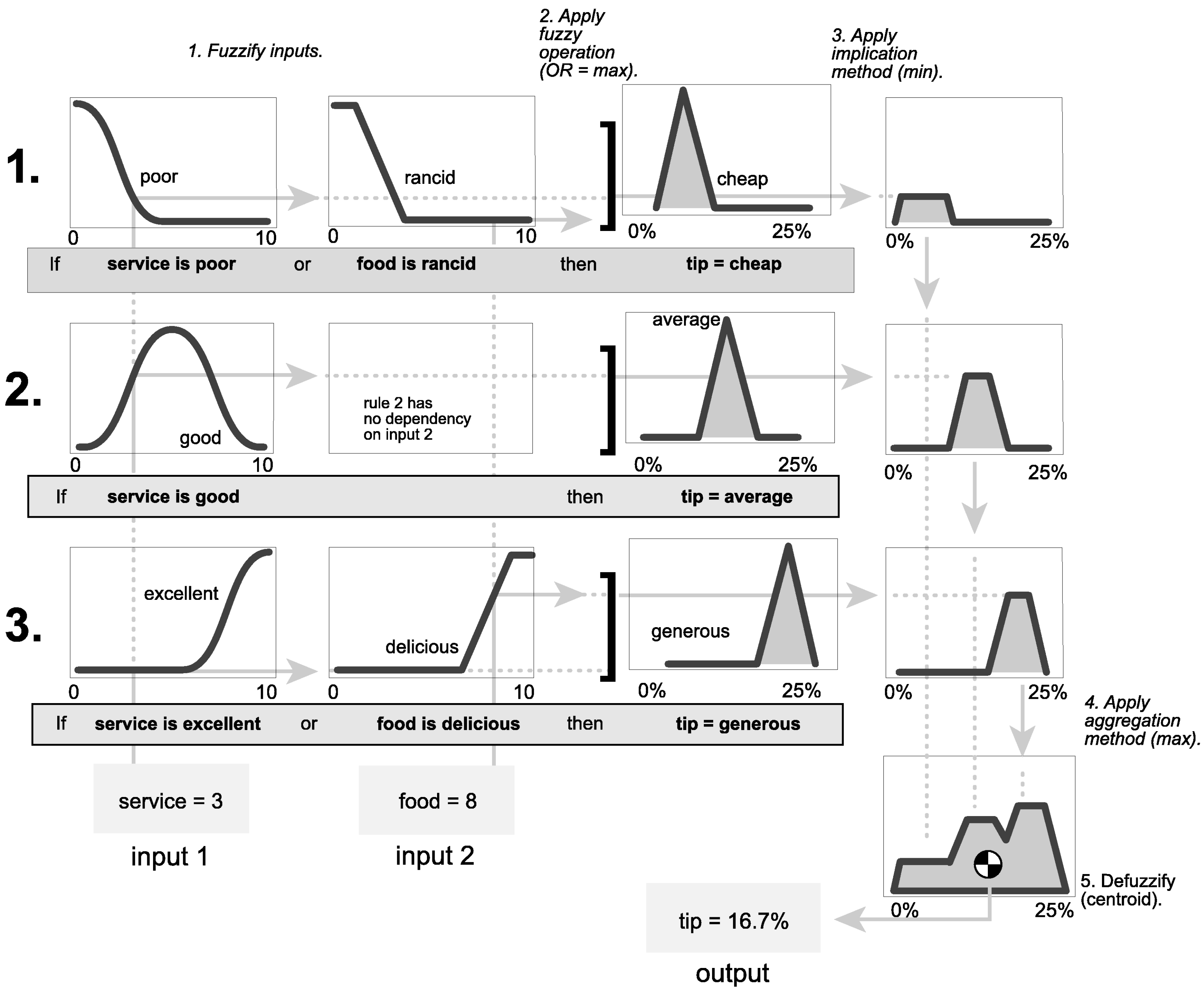
The fuzzy inference diagram is the composite of all the smaller diagrams weve been looking at so far in this section.
It simultaneously displays all parts of the fuzzy inference process weve examined.
Information flows through the fuzzy inference diagram as shown below.

Interpreting the fuzzy inference diagram



The flow proceeds up from the inputs in the lower left, then across each row, or rule, and then down the rule outputs to finish in the lower right.

This is a very compact way of showing everything at once, from linguistic variable fuzzification all the way through defuzzification of the aggregate output.


1. Fuzzify inputs.
2. Apply fuzzy operation (OR = max).
3. Apply implication method (min).
1.
poor
rancid
cheap
0
10
0%
25%
If service is poor or food is rancid then tip = cheap
2.
good
rule 2 has no dependency on input 2
average
If service is good then tip = average
3.
excellent
delicious
generous
If service is excellent or food is delicious then tip = generous
4. Apply aggregation method (max).
5. Defuzzify (centroid).
service = 3
input 1
food = 8
input 2
tip = 16.7%
output

- Solution to the tipping problem, can be presented as a 3-D surface equivalent to the following mapping:

$$\text{tip} = f(\text{service}, \text{food})$$