UNIT III KNOWLEDGE INFERENCE

Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network-Dempster Shafer theory.

PRODUCTION BASED SYSTEM

Production systems are also known as **rule based system**.

A system whose knowledge base is represented as a set of rules and facts is called rule based system. A rule based system consists of a collection of IF-THEN rules, a collection of facts, and some interpreter controlling the application of the rules given the facts.

Production rules

- IF-THEN expressions
- **IF** some condition(s) exists **THEN** perform some action(s)

Condition-action pair

- **Condition:** pattern that determines when a rule may be applied to problem instance
- **Action:** defines associated problem solving step

Antecedent – Consequent

- IF <antecedent> THEN <consequent>

When the antecedent part is NULL, the rule becomes fact.

Rule can have multiple antecedents

Conjunction AND

IF <antecedent$_0$> AND <antecedent$_1$> … AND <antecedent$_n$>
THEN <consequent>

Disjunction OR

IF <antecedent$_0$> OR <antecedent$_1$> … OR <antecedent$_n$>
THEN <consequent>

Combination of both

IF <antecedent0> AND <antecedent1> … AND <antecedentn>
OR <antecedent$_0$> OR <antecedent$_1$> … OR <antecedent$_n$>
THEN <consequent>

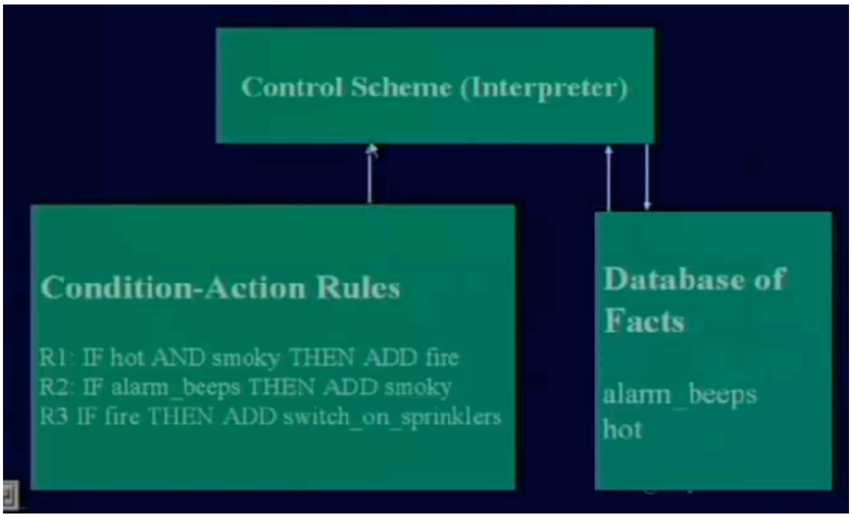
Consequents can also have multiple clauses

IF <antecedent> THEN <consequent$_0$>, <consequent$_1$>,
…<consequent$_{n-1}$>, <consequent$_n$>

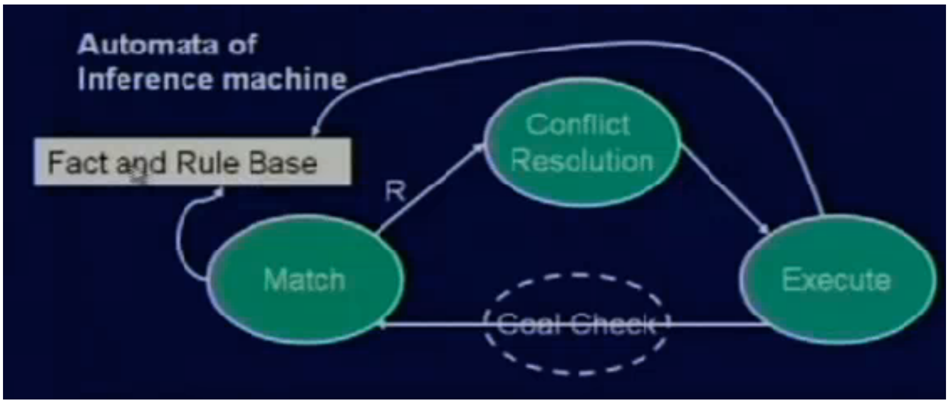
Triggered and fired rules

A rule is **triggered** when all the antecedents evaluate to true.

A rule is fired when the action stated in the consequent part or the inference related to the consequent part is inferred or taken.



Production based system



Inference machine is a machine that implements strategies to utilize the knowledge base and derive new conclusions from it.

Match

Inference machine is comparing the fact base and the rule base and find some rules are matching. These set of rules are passed to the next phase, conflict resolution.

Conflict resolution phase

- *Rule* 1:IF the 'traffic light' is green THEN the action is go
- *Rule* 2:IF the 'traffic light' is red THEN the action is stop
- *Rule* 3:IF the traffic light' is red THEN the action is go

We have two rules, *Rule* 2 and *Rule* 3, with the same IF part. Thus both of them can be set to fire when the condition part is satisfied. These rules represent a conflict set. The inference engine must determine which rule to fire from such a set. A method for choosing a rule to fire when more than one rule can be fired in a given cycle is called **conflict resolution**.

The conflict resolution mechanisms are

- Rule Ordering
- Recency
- Specificity
- Refraction
- Once a rule

i) Recency

- Rules which use more recent facts are preferred.
- Working memory elements are time tagged indicating at what cycle each fact was added to working memory.
- Focuses on single line of reasoning

ii) Specificity

- Rules which have greater number of conditions are therefore more difficult to satisfy, are preferred to more general rules with fewer conditions.
- More specific rules are better because they take more of the data in to account.

iii) Refraction

- A rule should not be allowed to fire more than once on the same data.
- The executed rules are discarded from the conflict set.
- Prevents undesired loops

iv) Rule ordering

- Choose the first rule in the text ordered top to bottom

Alternative to conflict resolution

Meta rules reason about which rules should be considered for firing. They direct reasoning rather than actually performing reasoning. Meta knowledge is knowledge about knowledge to guide search.

If conflict set contains any rule (c,a) such that c="animal is mammal" THEN fire(c,a). This example says meta knowledge encodes knowledge about how to guide search for solution.

Execute phase

The selected rule is fired in this phase. This phase also checks whether the goal is reached. After every execution new fact will be added to the fact base.

Limitations of Rule-Based Representations

- Can be difficult to create
  - the "knowledge engineering" problem
- Can be difficult to maintain
  - in large rule-bases, adding a rule can cause many unforeseen interactions and effects => difficult to debug
- Many types of knowledge are not easily represented by rules
  - uncertain knowledge: "if it is cold it will probably rain"
  - information which changes over time
  - procedural information (e.g. a sequence of tests to diagnose a disease)
  -

FRAME BASED SYSTEM

A frame is a data structure with typical knowledge about a particular object or concept. Frames, first proposed by **Marvin Minsky** in the 1970s.

Each frame has its own name and a set of **attributes** associated with it.

Frame - Person

Name
Weight
Height attributes or slots
Age

Each attribute or slot has a value attached to it. Frames provide a natural way for the structured and brief representation of knowledge. Frames are an application of **object-oriented programming** for expert systems.

When an object is created in an object-oriented programming language, we first assign a name to the object, and then determine a set of attributes to describe the object's characteristics, and at last write procedures to specify the object's behaviour. A knowledge engineer refers object as a **frame.**

Frames as a knowledge representation technique

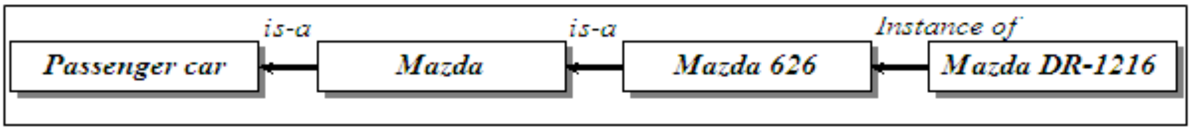
- The concept of a frame is defined by a collection of **slots**. Each slot describes a particular attribute or operation of the frame.
- Slots are used to store values. A slot may contain a default value or a pointer to another frame, a set of rules or procedure by which the slot value is obtained.

Typical information included in a slot

- **Frame name.**
- **Relationship of the frame to the other frames.** The frame *IBM Aptiva S35* might be a member of the class *Computer*, which in turn might belong to the class *Hardware*.
- **Slot value.** A slot value can be symbolic, numeric or Boolean. For example, the slot *Name* has symbolic values, and the slot *Age* numeric values
- **Default slot value.** The default value is taken to be true when no evidence to the contrary has been found. For example, a car frame might have four wheels and a chair frame four legs as default values in the corresponding slots.
- **Range of the slot value.** The range of the slot value determines whether a particular object complies with the stereotype requirements defined by the frame. For example, the cost of a computer might be specified between $750 and $1500.

- **Procedural information.** A slot can have a procedure attached to it, which is executed if the slot value is changed or needed.

Class inheritance in frame-based systems



CLASS:	*Passenger car*	
[C]	***Engine type***	
	In-line 4 cylinder:	
	V6:	
[N]	***Horsepower:***	
[C]	***Drivetrain type***	
	Rear wheel drive:	
	Front wheel drive:	
	Four wheel drive:	
[C]	***Transmission type***	
	5-speed manual:	
	4-speed automatic:	
[N]	***Fuel consumption (mpg):***	
[N]	***Seating capacity:***	

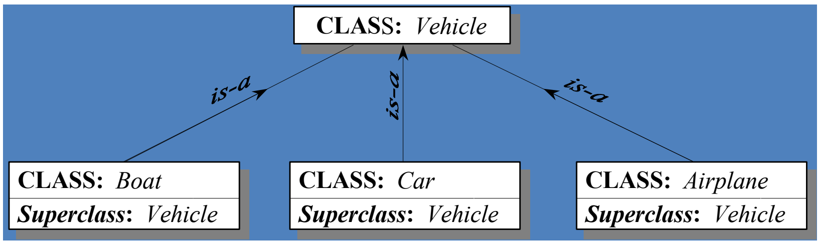
CLASS:	*Mazda*	
Superclass:	*Passenger car*	
[C]	***Engine type***	
	In-line 4 cylinder:	
	V6:	
[N]	***Horsepower:***	
[C]	***Drivetrain type***	
	Rear wheel drive:	
	Front wheel drive:	
	Four wheel drive:	
[C]	***Transmission type***	
	5-speed manual:	
	4-speed automatic:	
[N]	***Fuel consumption (mpg):***	
[N]	***Seating capacity:***	
[Str]	***Country of manufacture:***	*Japan*

CLASS:	*Mazda 626*	
Superclass:	*Mazda*	
[C]	***Engine type***	
	In-line 4 cylinder:	
	V6:	
[N]	***Horsepower:***	*125*
[C]	***Drivetrain type***	
	Rear wheel drive:	
	Front wheel drive:	
	Four wheel drive:	
[C]	***Transmission type***	
	5-speed manual:	
	4-speed automatic:	
[N]	***Fuel consumption (mpg):***	*22*
[N]	***Seating capacity:***	*5*
[Str]	***Country of manufacture:***	*Japan*
[Str]	***Model:***	
[C]	***Colour***	
	Glacier White:	
	Sage Green Metallic:	
	Slate Blue Metallic:	
	Black Onyx Clearcoat:	
[Str]	***Owner:***	

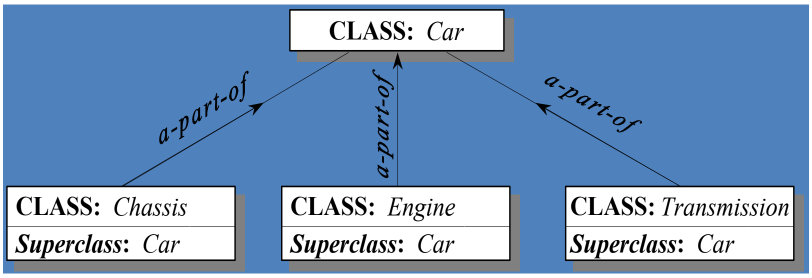
INSTANCE:	*Mazda DR-1216*	
Class:	*Mazda 626*	
[C]	***Engine type***	
	In-line 4 cylinder:	TRUE
	V6:	FALSE
[N]	***Horsepower:***	*125*
[C]	***Drivetrain type***	
	Rear wheel drive:	FALSE
	Front wheel drive:	TRUE
	Four wheel drive:	FALSE
[C]	***Transmission type***	
	5-speed manual:	FALSE
	4-speed automatic:	TRUE
[N]	***Fuel consumption (mpg):***	*28*
[N]	***Seating capacity:***	*5*
[Str]	***Country of manufacture:***	*Japan*
[Str]	***Model:***	*DX*
[C]	***Colour***	
	Glacier White:	FALSE
	Sage Green Metallic:	TRUE
	Slate Blue Metallic:	FALSE
	Black Onyx Clearcoat:	FALSE
[Str]	***Owner:***	*Mr Black*

Relationships among objects

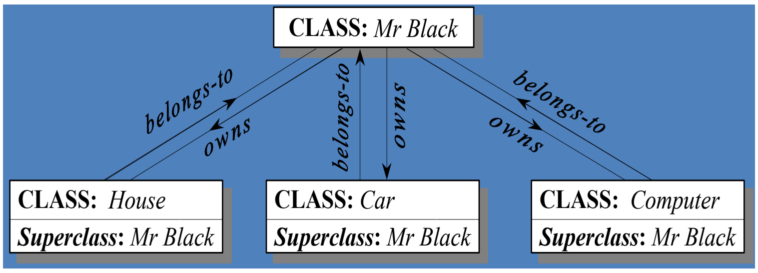
- **Generalisation** denotes *a-kind-of* or *is-a* relationship between superclass and its subclasses. Each subclass inherits all features of the superclass.



- **Aggregation** is ***a-part-of*** or ***part-whole*** relationship in which several subclasses representing *components* are associated with a superclass representing a *whole*.
  For example, an engine is *a part of* a car.



- **Association** describes some semantic relationship between different classes which are unrelated otherwise. Classes *House*, *Car* and *Computer* are mutually independent, but they are linked with the frame *Mr Black* .



- Most frame-based expert systems use two types of methods:
  - **WHEN CHANGED**
  - **WHEN NEEDED**.

A WHEN CHANGED method is executed immediately when the value of its attribute changes.
A WHEN NEEDED method is used to obtain the attribute value only when it is needed.

Interaction of frames and rules

Most frame-based expert systems allow us to use a set of rules to evaluate information contained in frames.

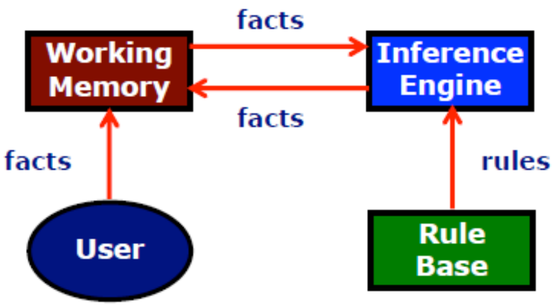
INFERENCE MECHANISM

Given a set of rules, there are essentially two ways to generate new knowledge

- Forward chaining
- Backward chaining

FORWARD CHAINING

Forward chaining is also known as data driven approach. Forward chaining starts with the facts and see what rules apply.



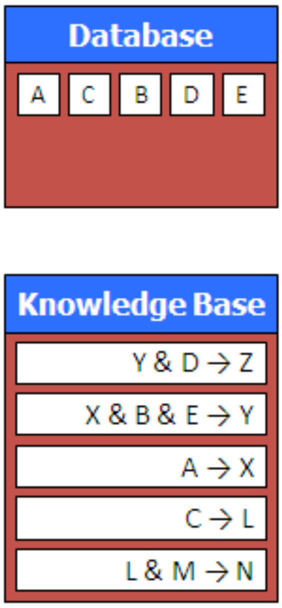
Facts are held in the working memory (ie., contains the current state of the world). Rules represent the action to be taken when specified facts occur in the working memory. The actions involve adding or deleting facts from the working memory.

Algorithm: Forward chaining

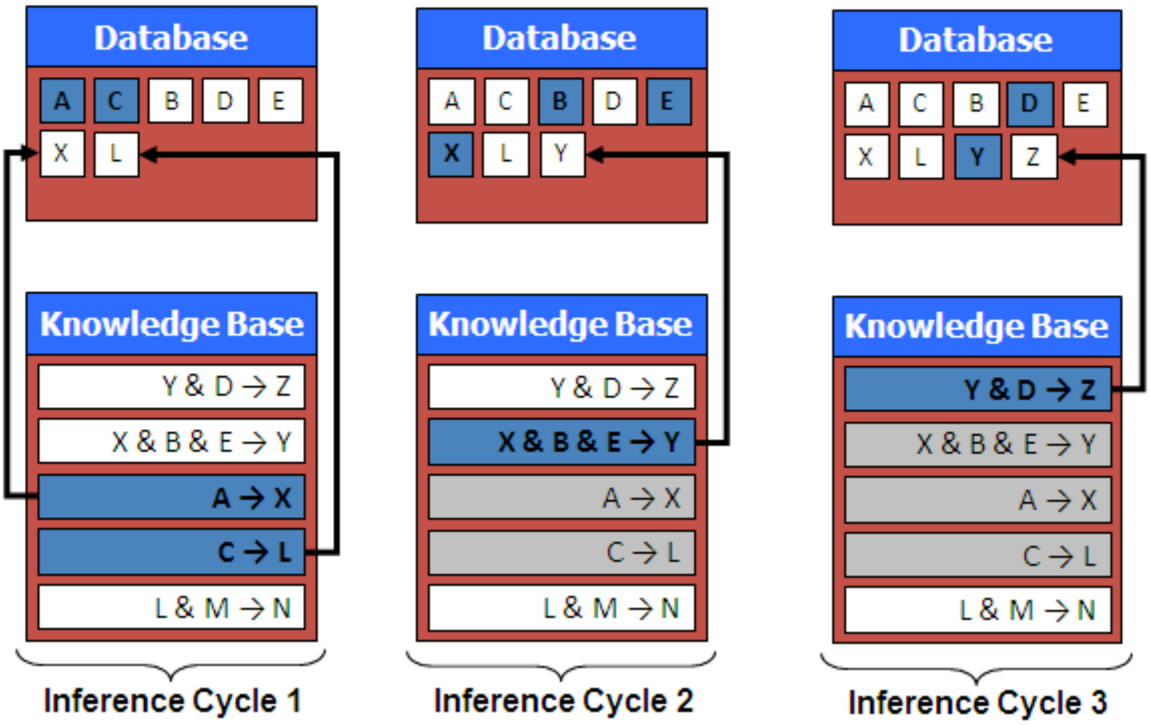
i) Collect the rules whose conditions match facts in working memory.
ii) If more than one rule matches then
(a) Use conflict resolution strategy to select the rules.
iii) Do the actions indicated by the rules. ie add facts to working memory or delete facts from working memory.
iv) Repeat these steps until the goal is reached or no condition match found.\

Example
Rule R1: IF hot and smoky THEN fire
Rule R2: IF alarm_beeps THEN smoky
Rule R3: IF fire THEN switch_on_sprinklers
Fact F1: alarm_beeps (Given)
Fact F2: hot (Given)

Inference
Rule R1: IF hot and smoky THEN fire
Rule R2: IF alarm_beeps THEN smoky
Rule R3: IF fire THEN switch_on_sprinklers
Fact F1: alarm_beeps (Given)
Fact F2: hot (Given)
Fact F3: smoky (Added)
Fact F4: fire (Added)
Fact F5: switch_on_sprinklers (Added)



Suppose that we know the facts A, B, C, D, E and the rules shown in the knowledge base to the left What facts can we infer from this?



After inferring facts X, L, Y and Z there are no more rules that can be fired.

Properties of forward chaining

- All rules which can fire do fire.
- Can be inefficient, which lead to spurious rule firing, unfocussed problem solving.
- Set of rules that can fire known as conflict set.
- Decision about which rule to fire is conflict resolution.

BACKWARD CHAINING

This is also called goal driven approach. A desired goal is placed in working memory, inference cycle attempts to find evidence to prove it.

The knowledge base (rule base) is searched for rules that might lead to goal. If condition of such rule matches fact in working memory then rule is fired and goal is proved.

Backward chaining means reasoning from goals to facts. Rules and facts are processed using backward chaining interpreter.

Algorithm: Backward Chaining

i) Prove goal G.

ii) if G is the initial fact, it is proven.

iii) Otherwise, find a rule which can be used to conclude G, and try to prove each of the rules conditions.

Example 1

Rule R1: IF hot and smoky THEN fire

Rule R2: IF alarm_beeps THEN smoky

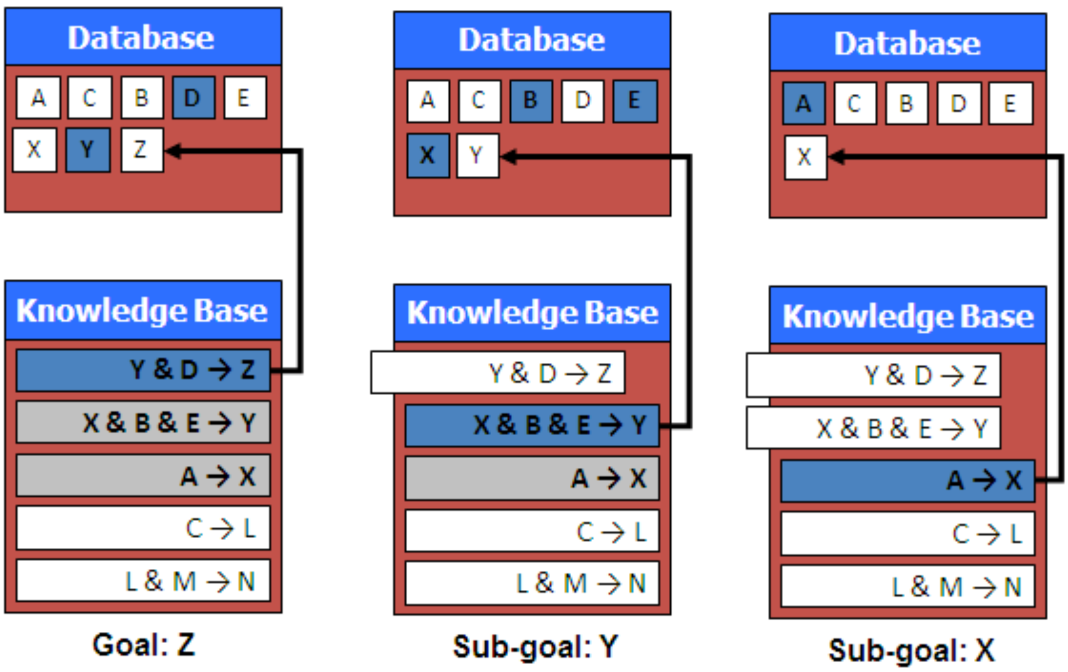
Rule R3: IF fire THEN switch_on_sprinklers

Fact F1: alarm_beeps (Given)

Fact F2: hot (Given)

Goal: Should I switch_on_sprinklers

Suppose that we know the facts A, B, C, D, E and the rules shown in the knowledge base to the left. Can we infer the fact Z?



Backward chaining inferred Z from the facts and rules that were available.

Advantages

- Most efficient to infer one particular fact.
- User may be asked to input additional facts

FUZZY REASONING

Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO. In fuzzy logic, the degree of truth is between 0 and 1.

Example: William is smart (0.8 truth)

The fuzzy logic works on the levels of possibilities of input to achieve the definite output.

Fuzzy logic is useful for commercial and practical purposes.

- It can control machines and consumer products.
- It may not give accurate reasoning, but acceptable reasoning.
- Fuzzy logic helps to deal with the uncertainty in engineering.

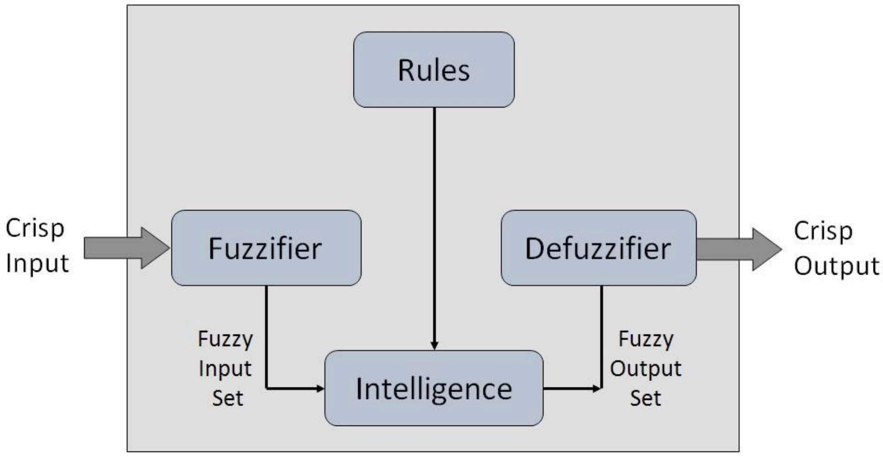
Fuzzy Logic Systems Architecture

It has four main parts as shown −

- **Fuzzification Module** − It transforms the system inputs, which are crisp numbers, into fuzzy sets. It splits the input signal into five steps such as −

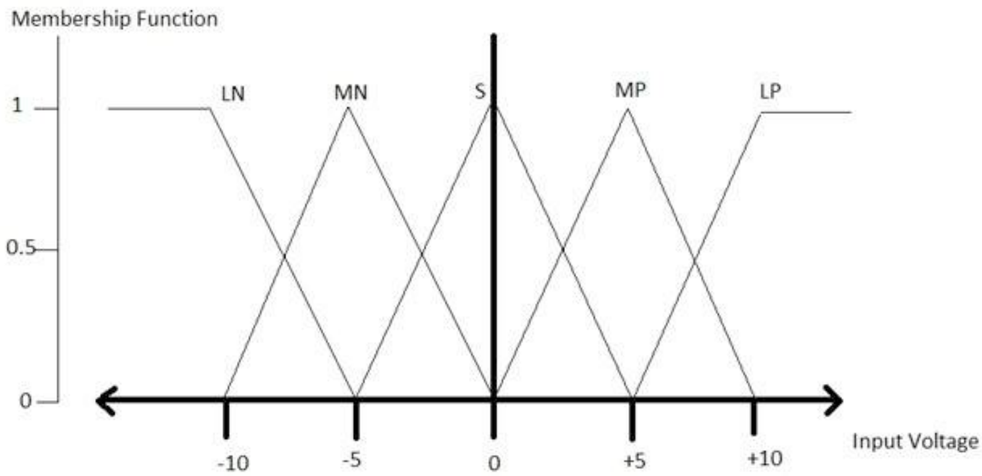
LP	x is Large Positive
MP	x is Medium Positive
S	x is Small
MN	x is Medium Negative
LN	x is Large Negative

- **Knowledge Base** − It stores IF-THEN rules provided by experts.
- **Inference Engine** − It simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.
- **Defuzzification Module** − It transforms the fuzzy set obtained by the inference engine into a crisp value.



A **membership function** for a fuzzy *set A* on the universe of discourse X is defined as $\mu A:X\rightarrow[0,1]$.Here, each element of *X* is mapped to a value between 0 and 1. It is called membership value or degree of membership.

All membership functions for **LP, MP, S, MN,** and **LN** are shown as below −



Here, the input to 5-level fuzzifier varies from -10 volts to +10 volts. Hence the corresponding output also changes.

Example of a Fuzzy Logic System

Let us consider an air conditioning system with 5-lvel fuzzy logic system. This system adjusts the temperature of air conditioner by comparing the room temperature and the target temperature value.

Algorithm

- Define linguistic variables and terms.
- Construct membership functions for them.
- Construct knowledge base of rules.
- Convert crisp data into fuzzy data sets using membership functions. (fuzzification)
- Evaluate rules in the rule base. (interface engine)
- Combine results from each rule. (interface engine)
- Convert output data into non-fuzzy values. (defuzzification)

Logic Development

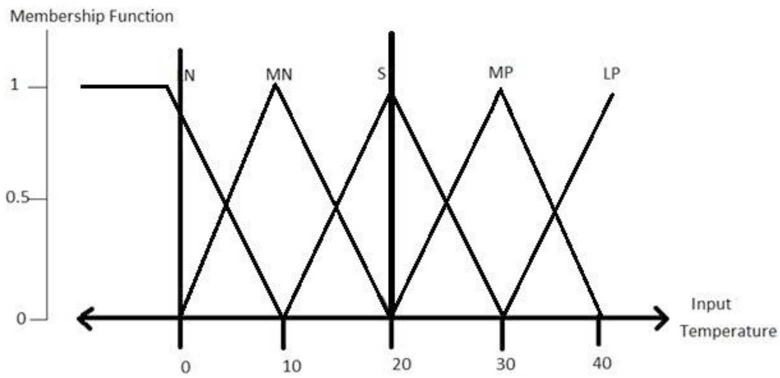
Step 1: Define linguistic variables and terms

Linguistic variables are input and output variables in the form of simple words or sentences. For room temperature, cold, warm, hot, etc., are linguistic terms.

Temperature (t) = {very-cold, cold, warm, very-warm, hot}

Step 2: Construct membership functions for them

The membership functions of temperature variable are as shown −



Step3: Construct knowledge base rules

Create a matrix of room temperature values versus target temperature values that an air conditioning system is expected to provide.

RoomTemp. /Target	Very_Cold	Cold	Warm	Hot	Very_Hot
Very_Cold	No_Change	Heat	Heat	Heat	Heat
Cold	Cool	No_Change	Heat	Heat	Heat
Warm	Cool	Cool	No_Change	Heat	Heat
Hot	Cool	Cool	Cool	No_Change	Heat
Very_Hot	Cool	Cool	Cool	Cool	No_Change

Build a set of rules into the knowledge base in the form of IF-THEN-ELSE structures.

Sr. No.	Condition	Action
1	IF temperature=(Cold OR Very_Cold) AND target=Warm THEN	Heat
2	IF temperature=(Hot OR Very_Hot) AND target=Warm THEN	Cool

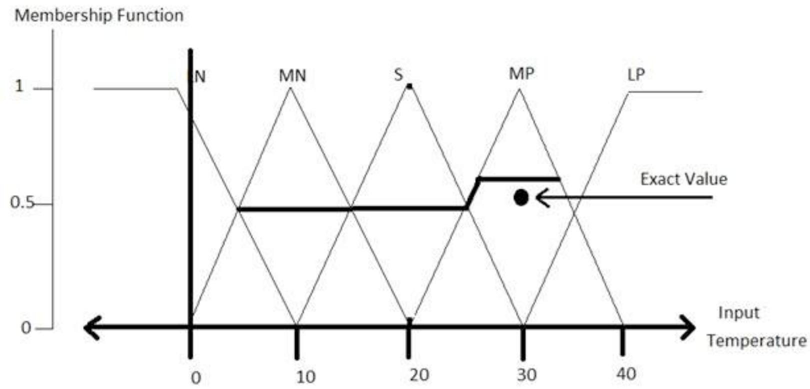
3	IF (temperature=Warm) AND (target=Warm) THEN	No_Change

Step 4: Obtain fuzzy value

Fuzzy set operations perform evaluation of rules. The operations used for OR and AND are Max and Min respectively. Combine all results of evaluation to form a final result. This result is a fuzzy value.

Step 5: Perform defuzzification

Defuzzification is then performed according to membership function for output variable.



Application Areas of Fuzzy Logic

Automotive Systems
- Automatic Gearboxes
- Four-Wheel Steering
- Vehicle environment control

Consumer Electronic Goods
- Hi-Fi Systems
- Photocopiers
- Still and Video Cameras
- Television

Domestic Goods
- Microwave Ovens
- Refrigerators
- Toasters
- Vacuum Cleaners
- Washing Machines

Environment Control
- Air Conditioners/Dryers/Heaters
- Humidifiers

Advantages of Fuzzy Logic System
- Mathematical concepts within fuzzy reasoning are very simple.
- Able to modify Fuzzy Logic System by just adding or deleting rules due to flexibility of fuzzy logic.
- Fuzzy logic Systems can take imprecise, distorted, noisy input information.
- FLSs are easy to construct and understand.
- Fuzzy logic is a solution to complex problems in all fields of life, including medicine, as it resembles human reasoning and decision making.

Disadvantages of Fuzzy Logic System
- There is no systematic approach to fuzzy system designing.
- They are understandable only when simple.
- They are suitable for the problems which do not need high accuracy.

CERTAINTY FACTORS

A certainty factor (CF) is a numerical value that expresses a degree of subjective belief that a particular item is true. The item may be a fact or a rule. When probabilities are used attention must be paid to the underlying assumptions and probability distributions in order to show validity. Bayes' rule can be used to combine probability measures.

Suppose that a certainty is defined to be a real number between -1.0 and +1.0, where 1.0 represents complete certainty that an item is true and -1.0 represents complete certainty that an item is false. Here a CF of 0.0 indicates that no information is available about either the truth or the falsity of an item. Hence positive values indicate a degree of belief or evidence that an item is true, and negative values indicate the opposite belief. Moreover it is common to select a positive number that represents a minimum threshold of belief in the truth of an item. For example, 0.2 is a commonly chosen threshold value

Certainty Factors and Facts and Rules

CFs with facts

padre(John, Mary) with CF .90

CFs with rules

(if (sneezes X) then (has_cold X)) with CF 0.7

– where the CF measures our belief in the conclusion given the premise is observed.

CFs are calculated using two other measures:

1. MB(H, E) – Measure of Belief: value between 0 and 1 representing the degree to which belief in the hypothesis H is supported by observing evidence E.

$$MB(H,E) = \begin{cases} 1 & \text{si } p(H) = 1 \\ \frac{p(H|E)-p(H)}{1-p(H)} & \text{si } p(H) < 1 \end{cases}$$

It is intended to capture the degree to which the evidence increases probability: p(H|E)-p(H) in proportion to the maximum possible increase in probability: 1-p(H)

2. MD(H, E) – Measure of Disbelief: value between 0 and 1 representing the degree to which disbelief in the hypothesis H is supported by observing evidence E.

$$MD(H,E) = \begin{cases} 1 & \text{si } p(H) = 0 \\ \frac{p(H)-p(H|E)}{p(H)} & \text{si } p(H) > 0 \end{cases}$$

CF is calculated in terms of the difference between MB and MD:

$$CF(H,E) = MB(H,E) - MD(H,E)$$

Combining Certainty Factors

• Multiple sources of evidence produce CFs for the same fact.
• For instance, two rules may provide evidence for the same conclusion:
– if P1 Then C CF=0.8
– if P2 Then C CF=0.7
• We need to know how to combine the CFs in such cases
• If two rules both support a hypothesis, then that should increase our belief in the hypothesis.

To combine certainty factors, use an incremental approach as with Bayes

$$CF(H, E \wedge E') = \begin{cases} X + Y(1-X) & \text{if } X, Y > 0 \\ X + Y(1+X) & \text{if } X, Y < 0 \\ \frac{X+Y}{1-min(|X|,|Y|)} & \text{if } sign(X) \neq sign(Y) \end{cases}$$

Rules with uncertain evidence
When a rule has a single premise, the certainty of the conclusion is the PRODUCT of the certainty of the premise multiplied by the certainty of the rule:

CF(C) = CF(P) * CF(R1)

Rules with uncertain evidence: negative evidence
A rule is only applicable if you believe the premise to be true

CF(C) = CF(P) * CF(RULE) if CF(P)>0
= 0 otherwise

Rules with uncertain evidence: more than one premise
If a rule has more than one premise:
IF P1&P2&P3 THEN C
• We find the CF of the set of premises – WHICH is just the MIN
CFPs = MIN(CF(P1), CF(P2), CF(P3))
CF(C) = CFPs * CF(RULE) if CFPs>0
= 0 otherwise

BAYESIAN THEORY

It is the basis of uncertain reasoning where the results are unpredictable.
Bayes Rule

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

P(h)- prior probability of hypothesis h
P(D)prior probability of data D, evident
P(h|D)-posterior probability
P(D|h)- likelihood of D given h
Axioms of probability

1. All probabilities are between 0 and 1 ie$0 \leq P(A) \leq 1$
2. P(True)=1 and P(false)=0
3. $P(A \lor B)=P(A)+P(B)-P(A \land B)$

BAYESIAN NETWORK

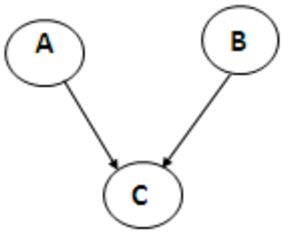
- A Bayesian network is a probabilistic graphical model that represents a set of variables and their probabilistic independencies. Otherwise known as Bayes net, Bayesian belief Network or simply Belief Networks. A Bayesian network specifies a joint distribution in a structured form. It represent dependencies and independence via a directed graph. Networks of concepts linked with conditional probabilities.

- Bayesian network consists of
  - Nodes = random variables
  - Edges = direct dependence
- Directed edges => direct dependence
- Absence of an edge => conditional independence
- Requires that graph is acyclic (no directed cycles)
- 2 components to a Bayesian network
  - The graph structure (conditional independence assumptions)
  - The numerical probabilities (for each variable given its parents)

For eg, evidence says that lab produces 98% accurate results. It means that a person X has 98% malaria or 2% of not having malaria. This factor is called uncertainty factor. This is the reason that we go for Bayesian theory. Bayesian theory is also known as probability learning.

The probabilities are numeric values between 0 and 1 that represent uncertainties.

i) Simple Bayesian network

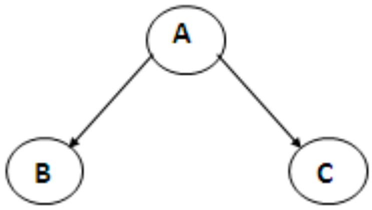


$$p(A,B,C) = p(C|A,B)p(A)p(B)$$

ii) 3-way Bayesian network (Marginal Independence)

$$p(A,B,C) = p(A)\ p(B)\ p(C)$$

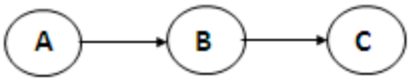
iii) 3-way Bayesian network (Conditionally independent effects)



$$p(A,B,C) = p(B|A)p(C|A)p(A)$$

B and C are conditionally independent Given A

iv) 3-way Bayesian network (Markov dependence)



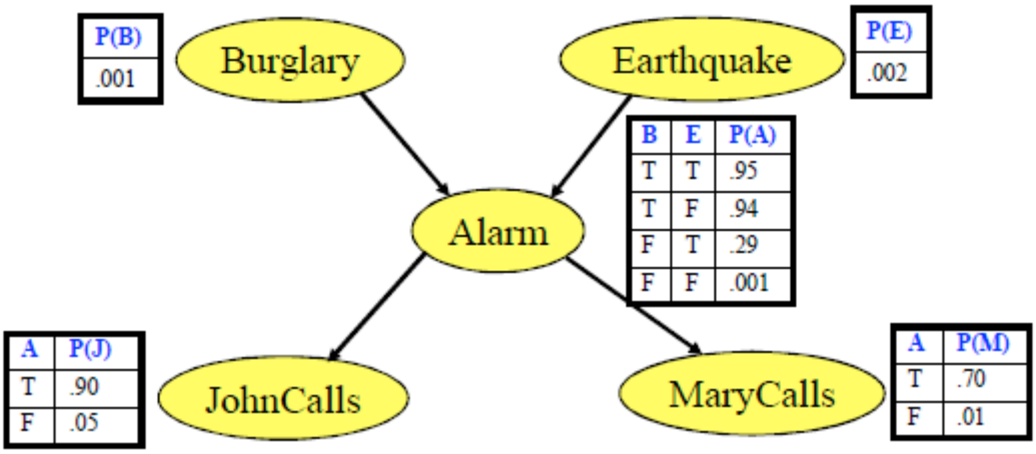
$$p(A,B,C) = p(C|B)\ p(B|A)p(A)$$

<u>Problem 1</u>

You have a new burglar alarm installed. It is reliable about detecting burglary, but responds to minor earth quakes. Two neighbors (John, Mary) promise to call you at work when they hear the alarm. John always calls when hears alarm, but confuses with phone ringing. Mary likes lod nusic and sometimes misses alarm. Find the probability of the event that the alarm has sounded but neither a burglary nor an earth quake has occurred and both Mary and John call.

Consider 5 binary variables

- B=Burglary occurs at your house
- E=Earth quake occurs at your home
- A=Alarm goes off
- J=John calls to report alarm
- M=Mary calls to report the alarm



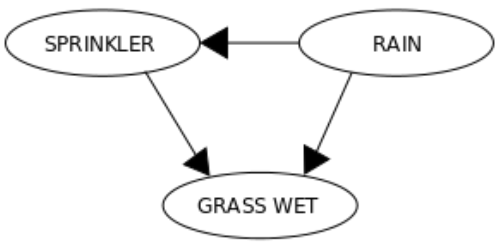
Probability of the event that the alarm has sounded but neither a burglary nor an earth quake has occurred and both Mary and John call

$$P(J,M,A,E,B)=P(J|A).P(M|A).P(A|E,B).P(E).P(B)$$
$$=0.90*0.70*0.001*0.99*0.998$$
$$=0.00062$$

<u>Problem 2</u>

Rain influences sprinkler usage. Rain and sprinkler influences whether grass is wet or not. What is the probability that rain gives grass wet?

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Solution

Let S= Sprinkler

R=Rain

G=Grass wet

$$P(G,S,R)=P(G|S,R).P(S|R).P(R)$$
$$=0.99*0.01*0.2$$
$$=0.00198$$

<u>Problem 3</u>

Bayesian Classifier: Training Dataset

Class:

C1:buys_computer = ‘yes’

C2:buys_computer = ‘no’

Data sample

X = (age <=30, Income = medium, Student = yes Credit_rating = Fair)

age	income	student	credit_rating	/s_compu
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Solution

- **P(Ci):**

  P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- **Compute P(X|Ci) for each class**

  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

P(X|C_i) :

P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|C_i)*P(C_i) :

P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Problem 4

Did the patient have malignant tumour or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which a malignant tumour actually present, and a correct negative result in only 97% of the cases in which it is not present. Furthermore, o.oo8 of the entire population have this tumour.

Solution:

P(tumour)=0.008	P(¬tumour)=0.992
P(+|tumour)=0.98	P(-|tumour)=0.02
P(+|¬tumour)=0.03	P(-|¬tumour)=0.97

$$P(tumour)|+) = \frac{P(+|tumour)P(tumour)}{P(+)}$$

$$= \frac{0.98 * 0.008}{P(+)}$$

$$P(\neg tumour)|+) = \frac{P(+|\neg tumour)P(\neg tumour)}{P(+)}$$

$$= \frac{0.3 * 0.992}{P(+)}$$

$$\frac{0.98 * 0.008}{P(+)} + \frac{0.3 * 0.992}{P(+)} = 1$$

$$P(+) = 0.98 * 0.008 + 0.3 * 0.992 = 0.305$$

$$P(tumour)|+) = \frac{0.98 * 0.008}{0.305} = 0.025$$

$$P(\neg tumour)|+) = \frac{0.3 * 0.992}{0.305} = 0.975$$

The probability of not having tumour is high. So the person is not having malignant tumour.

DEMPSTER - SHAFER THEORY

- This means that it is possible to believe that something could be both true and false to some degree

Dempster-Shafer theory is an approach to combining evidence. Dempster (1967) developed means for combining degrees of belief derived from independent items of evidence.

Each fact has a degree of support, between 0 and 1:

– 0 No support for the fact

– 1 full support for the fact

• Differs from Bayesian approach in that:

– Belief in a fact and its negation need not sum to 1.

– Both values can be 0 (meaning no evidence for or against the fact)

Set of possible conclusions:

$\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$

– Θ is the set of possible conclusions to be drawn

– Each θ_i is mutually exclusive: at most one has to be true.

– Θ is Exhaustive: At least one θ_i has to be true.

Mass function m(A):

(where A is a member of the power set)= proportion of all evidence that supports this element of the power set.

Each m(A) is between 0 and 1.

• All m(A) sum to 1.

• m(Ø) is 0 - at least one must be true.

Belief in A:

The belief in an element A of the Power set is the sum of the masses of elements which are subsets of A (including A itself).

A={q1, q2, q3}

Bel(A) = m(q1)+m(q2)+m(q3) + m({q1, q2})+m({q2, q3})+m({q1, q3})+m({q1, q2, q3})

Plausibility of A: pl(A)

The plausibility of an element A, pl(A), is the sum of all the masses of the sets that intersect with the set A:

pl({B,J}) = m(B)+m(J)+m(B,J)+m(B,S)+m(J,S)+m(B,J,S)

Problem 1

- 4 people (B, J, S and K) are locked in a room when the lights go out.
- When the lights come on, K is dead, stabbed with a knife.
- Not suicide (stabbed in the back)
- No-one entered the room.
- Assume only one killer.

- ʘ= { B, J, S}
- P(ʘ) = (Ø, {B}, {J}, {S}, {B,J}, {B,S}, {J,S}, {B,J,S})

Mass function m(A):

- Detectives, after reviewing the crime-scene, assign mass probabilities to various elements of the power set:

Event	Mass
No-one is guilty	0
B is guilty	0.1
J is guilty	0.2
S is guilty	0.1
either B or J is guilty	0.1
either B or S is guilty	0.1
either S or J is guilty	0.3
One of the 3 is guilty	0.1

A	{B}	{J}	{S}	{B,J}	{B,S}	{J,S}	{B,J,S}
m(A)	0.1	0.2	0.1	0.1	0.1	0.3	0.1
bel(A)	0.1	0.2	0.1	0.4	0.3	0.6	1.0
pl(A)	0.4	0.7	0.6	0.9	0.8	0.9	1.0

The certainty associated with a given subset A is defined by the belief interval: [bel(A) pl(A)].
From this we observed that J is a killer.