Unit III

Knowledge Inference

KNOWLEDGE REPRESENTATION

Knowledge representation is probably, the most important ingredient for developing an AI. A representation is a layer between information accessible from outside world and high level thinking processes. Without knowledge representation it is impossible to identify what thinking processes are, mainly because representation itself is a substratum for a thought.

The subject of knowledge representation has been messaged for a couple of decades already. For many applications, specific domain knowledge is required. Instead of coding such knowledge into a system in a way that it can never be changed (hidden in the overall implementation), more flexible ways of representing knowledge and reasoning about it have been developed in the last 10 years.

The need of knowledge representation was felt as early as the idea to develop intelligent systems. With the hope that readers are well conversant with the fact by now, that intelligent requires possession of knowledge and that knowledge is acquired by us by various means and stored in the memory using some representation techniques. Putting in another way, knowledge representation is one of the many critical aspects, which are required for making a computer behave intelligently. Knowledge representation refers to the data structures techniques and organizing notations that are used in AI. These include semantic networks, frames, logic, production rules and conceptual graphs.

Properties for knowledge Representation

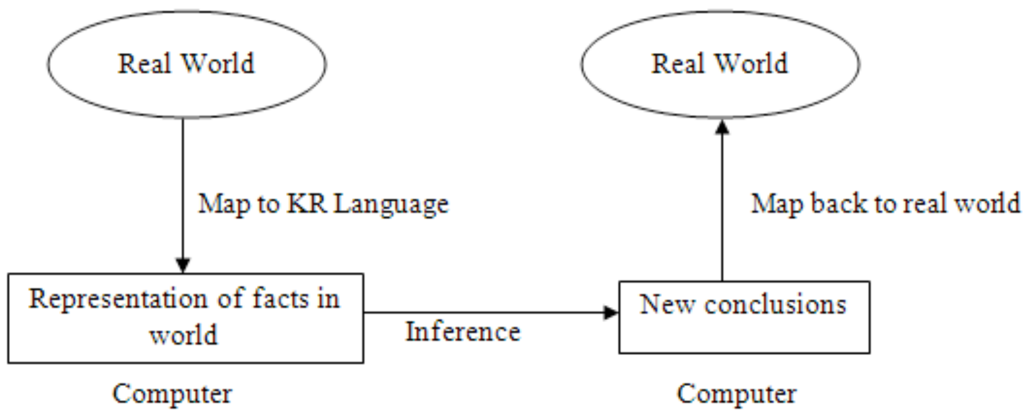
The following properties should be possessed by a knowledge representation system.

a. **Representational Adequacy:** It is the ability to represent the required knowledge.
b. **Inferential Adequacy:** It is the ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original.
c. **Inferential Efficiency:** The ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides.
d. **Acquisitional Efficiency:** The ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

Syntax and semantics for Knowledge Representation

Knowledge representation languages should have precise syntax and semantics. You must know exactly what an expression means in terms of objects in the real world. Suppose we have decided that "red 1" refers to a dark red colour, "car1" is my car, car2 is another. Syntax of language will tell you which of the following is legal: red1 (car1), red1 car1, car1 (red1), red1 (car1 & car2)?

Semantics of language tell you exactly what an expression means: for example, Pred (Arg) means that the property referred to by Pred applies to the object referred to by Arg. E.g., properly "dark red" applies to my car.



Types of Knowledge Representation

Knowledge can be represented in different ways. The structuring of knowledge and how designers might view it, as well as the type of structures used internally are considered. Different knowledge representation techniques are

a. Logic
b. Semantic Network
c. Frame
d. Conceptual Graphs
e. Conceptual Dependency
f. Script

FRAME Based System

A frame is a collection of attributes and associated values that describe some entity in the slot values. The slots may be of any size and type. Slots typically have names and values or subfields called facets. Facets may also have names and any number of values. A frame may have any number of slots, a slot may have any number of facets, each with any number of values. A slot contains information such as attribute value pairs, default values, condition for filling a slot, pointers to other related frames and procedures that are activated when needed for different purposes. Sometimes a frame describes an entity in some absolute sense, sometimes it represents the entity from a particular point of view. A single frame taken alone is rarely useful. We build frame systems out of collection of frames that are connected to each other by virtue of the fact that the value of an attribute of one frame may be another frame. Each frame should start with an open parenthesis and closed with a closed parenthesis.

Syntax of a frame

```
(<frame name>
(<slot1> (<facet1> <value 1>..........<value n1>)
      (<facet2> <value1>............<value n2>)
      .
      .
      .
      .
      .
      (<facet n> <value1>............ <value nn>))
(<slot 2> (<facet1> <value 1>......<value n1>)
      (<facet2><value2>...........<value n2>)
      .
      .
      ))
```

Let us consider the below examples.

1) Create a frame of the person Ram who is a doctor. He is of 40. His wife name is Sita. They have two children Babu and Gita. They live in 100 kps street in the city of Delhi in India. The zip code is 756005.

(Ram

(PROFESSION (VALUE

Doctor)) (AGE (VALUE 40))

(WIFE (VALUE Sita))

(CHILDREN (VALUE Bubu,

Gita)) (ADDRESS

(STREET (VALUE 100 kps))

(CITY(VALUE Delhi))

(COUNTRY(VALUE India))

(ZIP (VALUE 756005))))

2) Create a frame of the person Anand who is a chemistry professor in RD Women's College. His wife name is Sangita having two children Rupa and Shipa.

(Anand

(PROFESSION (VALUE Chemistry

Professor)) (ADDRESS (VALUE RD

Women's College)) (WIFE (VALUE

Sangita)) (CHILDREN(VALUE

RupaShipa)))

3) Create a frame of the person Akash who has a white maruti car of LX-400 Model. It has 5 doors. Its weight is 225kg, capacity is 8, and mileage is 15 km /lit.

(Akash

(CAR (VALUE Maruti))

(COLOUR (VALUE

White)) (MODEL

(VALUE LX-400))

(WEIGHT (VALUE

225kg)) (CAPACITY

(VALUE 8)) (MILAGE

(VALUE 15km/lit)))

The frames can be attached with another frame and can create a network of frames. The main task of action frame is to provide the facility for procedural attachment and help in reasoning process. Reasoning using frames is done by instantiation. Instantiation process begins, when the given situation is matched with frames that are already in existence. The reasoning process tries to match the current problem state with the frame slot and assigns them values.The valuesassigned to the slots depict a particular situation and by this, the reasoning process moves towards a goal. The reasoning process can be defined as filling slot values in frames.

Frames as Sets and Instances

The set theory is a good basis for understanding frame systems.

Each frame represents either a class (a set)or an instance (an element of class)

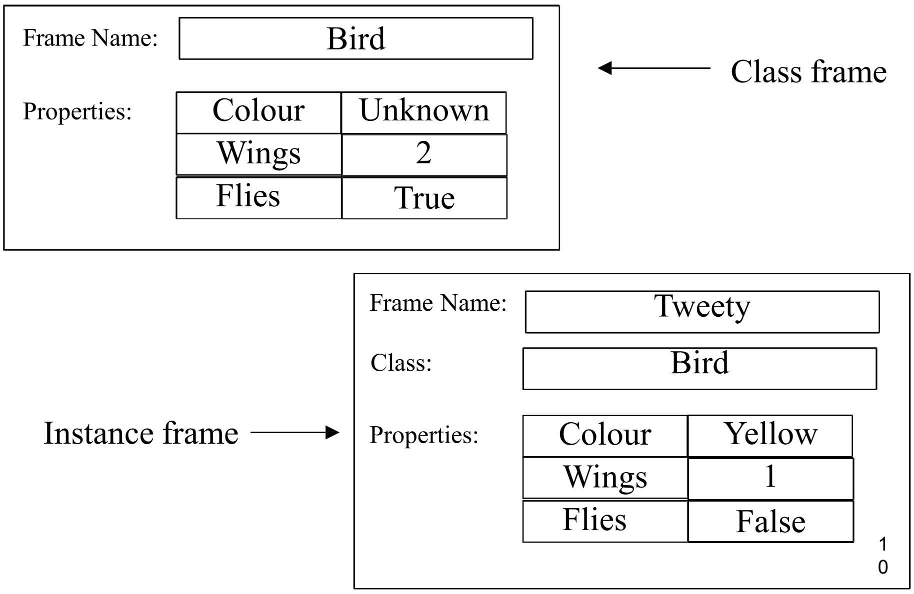
Considering the Cricket example – Person, Adult Male, Bowler, Team are all classes. Sachin and India are entities.

Both isa and instance relations have inverse attributes, which we call subclasses & all-instances.

As a class represents a set, there are 2 kinds of attributes that can be associated with it.

Its own attributes & Attributes that are to be inherited by each element of the set. Latter is represented by *)

Example of frames (1)

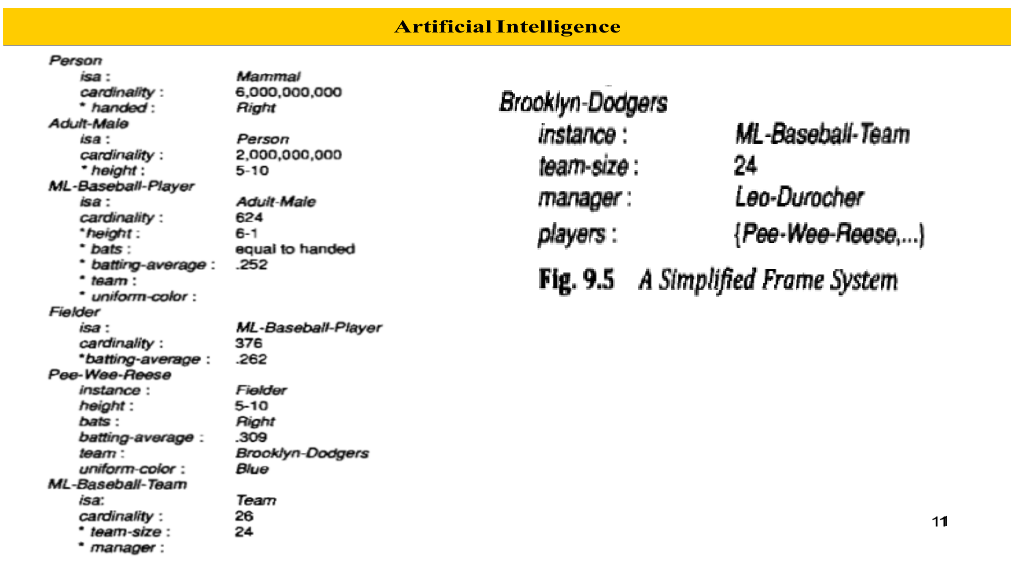


Sometimes, the difference between a set and an individual instance may not be clear. Example –Team India is an instance of class of Cricket Teams and can also be thought of as set of players.

Now the problem is if we represent Team India as a sub class of Cricket teams, then indian players automatically become part of all the teams, which is not true. We have to do something to stop this. Instead we can make Team India a sub class of class called Cricket Players.

To do this we need to differentiate between regular classes and meta classes.

Regular Classes are those whose elements are individual entities and metaclasses which are special classes whose elements are themselves classes.



The most basic meta class is the class CLASS. It represents the set of all classes.
All classes are instances of it, either directly or through one of its subclasses.

The class CLASS introduces the attribute cardinality, which is to be inherited by all instances of CLASS.

Other ways of Relating Classes to Each Other

We have discussed that

A class1 can be a subset of class2.

If Class2 is a meta class then Class1 can be an instance of Class2.

Another way is - mutually-disjoint-with relationship, which relates a class to one or more other classes that are guaranteed to have no elements in common with it.

Another one is – is-covered-by: - which relates a class to a set of subclasses, the union of which is equal to it.

If a class is-covered-by a set S of mutually disjoint classes, then S is called a partition of the class. Cardinality stands for number.

Slots as Full-Fledged Objects (Frames)

Till now we have used attributes as slots, but now we will like to represent attributes explicitly and describe their properties.

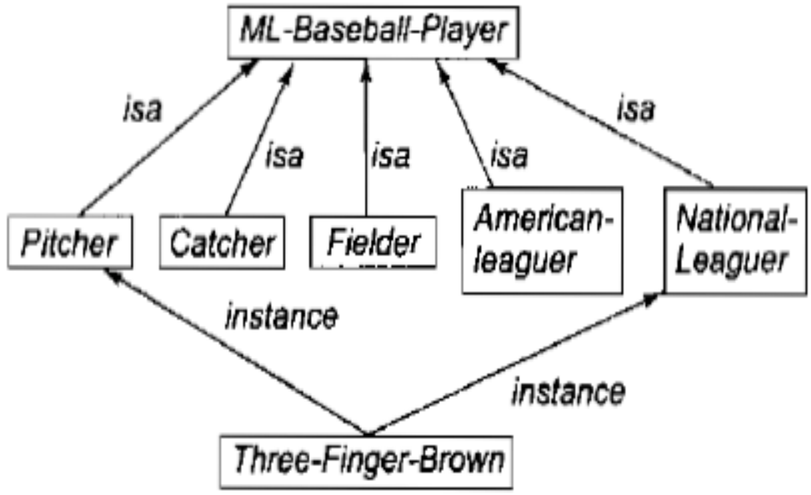
Some of the properties we would like to be able to represent and use in reasoning include:

1. The class to which the attribute can be attached?
2. Constraints on either the type or the value of the attribute.
3. A default value for the attribute.
4. Rules for inheriting values for the attribute.

To be able to represent these attributes of attributes, we need to describe attributes(slots) as frames. These frames will be organized into an isa hierarchy, just as any other frames are, and that hierarchy can then be used to support inheritance of values for attributes of slots.

Now let is formalize what is a slot.

A slot is a relation. It maps from elements of its domain (the classes for which it makes sense) to elements of its range (its possible values). A relation is a set of ordered pairs. Thus it makes sense to say that relation R1 is a subset of another R2.In that case R1 is a specialization of R2.Since a slot is a set, the set of all slots, which we will call SLOT, is a meta class. Its instances are slots, which may have sub slots.



ML-Baseball-Player	
is-covered-by :	{*Pitcher, Catcher, Fielder*}
	{*American-Leaguer, National-Leaguer*}
Pitcher	
isa :	*ML-Baseball-Player*
mutually-disjoint-with :	{*Catcher, Fielder*}
Catcher	
isa :	*ML-Baseball-Player*
mutually-disjoint-with:	{*Pitcher, Fielder*}
Fielder	
isa :	*ML-Baseball-Player*
mutually-disjoint-with :	{*Pitcher, Catcher*}
American-Leaguer	
isa :	*ML-Baseball-Player*
mutually-disjoint-with :	{*National-Leaguer*}
National-Leaguer	
isa :	*ML-Baseball-Player*
mutually-disjoint-with :	{*American-Leaguer*}
Three-Finger-Brown	
instance :	*Pitcher*
instance :	*National-Leaguer*

Fig. 9.8 *Representing Relationships among Classes*

Slots Values as Objects

slot "values" as objects. Let's take the following example –

John :

height : 72 Bill :

height :

To make a statement about the value of a slot without knowing what the value is,need to view the slot and its value as an object.

We will expand our representation language to allow the value of slot to be stated as either or both of :

A value of the type required by the slot.

A logical constraint on the value. The constraint may relate the slot's value to the values of other slots or to domain constants.

If we do this to the frames of Fig. 9.13, then we get the frames of Fig. 9.14. We again use the lambda notation as a way to pick up the name of the frame that is being described.

John
    height : 72; λx (*x.height* > *Bill.height*)
Bill
    height : λx (*x.height* < *John.height*)

Fig. 9.14 *Representing Slot-Values with Lambda Notation*

Advantages and Disadvantages of Different Knowledge Representation

Sl. No.	Scheme	Advantages	Disadvantages
1	Production rules	• Simple syntax • Easy to understand • Simple interpreter • Highly Modular • Easy to add or modify	• Hard to follow Hierarchies • Inefficient for large systems • Poor at representing structured descriptive knowledge.
2	Semantic	• Easy to follow hierarchy • Easy to trace associations • Flexible	• Meaning attached to nodes might be ambiguous • Exception handling is difficult • Difficult to program
3	Frame	• Expressive Power • Easy to set up slots for new properties and relations	• Difficult to program • Difficult for inference • Lack of inexpensive software

		procedures	
4	Script	• Ability to predict events • A single coherent interpretation may be build up from a collection of observations	• Less general than frames • May not be suitable to represent all kinds of knowledge
5	Formal Logic	• Facts asserted independently of use • Assurance that only valid consequence are asserted • Completeness	• Separation of representation and processing • Inefficient with large data sets • Very slow with large knowledge bases

Inference Engine

The inference engine is a generic control mechanism for navigating through and manipulating knowledge and deduce results in an organized manner.

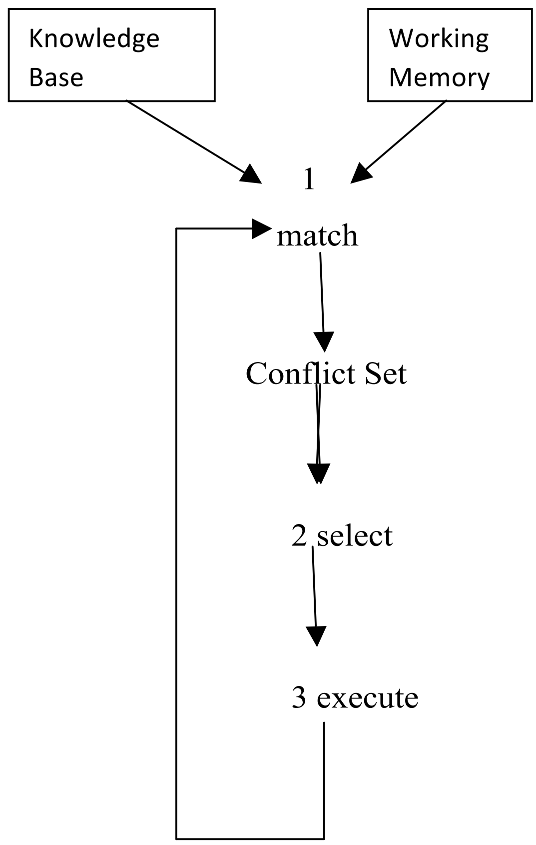
The inference engine's generic control mechanism applies the axiomatic (self-evident) knowledge present in the knowledge base to the task-specific data to arrive at some conclusion.

‡ Inference engine- the other key component of all expert systems.

‡ **Just** a knowledge base alone is not of much use if there are no facilities for navigating through and manipulating the knowledge to deduce something from knowledge base.

‡ A knowledge base is usually very large, it is necessary to have inferencing mechanisms that search through the database and deduce results in an organized manner.

The inference engine accepts user input queries and responses to questions through the I/O interface and uses this dynamic information together with the static knowledge (the rules and facts) stored in the knowledge base. The knowledge in the knowledge base is used to derive conclusions about the current case or situation as presented by the user's input. The inferring process is carried out in three stages.1)match 2) Select 3)execute. During the match the contents of working memory are compared to facts and rules contained in the knowledge base. When consistent matches are found, the corresponding rules are placed in the conflict set.To find an appropriate and consistent match substitutions may be required. Once all the matched rules have been added to the conflict set during a given cycle one of the rules is selected for execution. The criteria for selection may be most recent use, rule condition specificity, (the number of conjuncts on the left), or simply the smallest rule number. The selected rule is then executed and the right-hand side or action part of the rule is then carried out.



The Forward chaining, and Backward chaining are some of the techniques used for drawing inferences from the knowledge base.

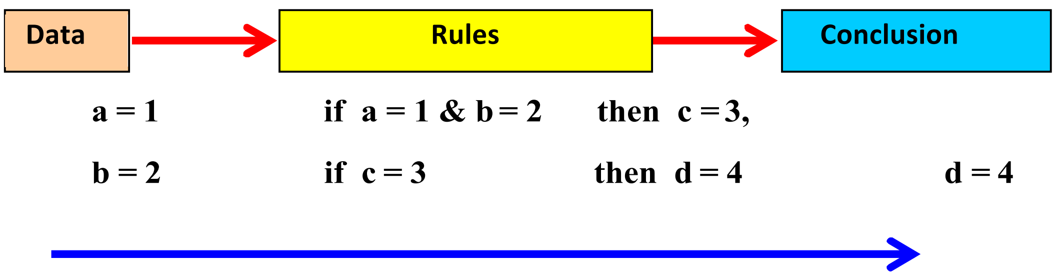
Forward Chaining Algorithm

Forward chaining is a techniques for drawing inferences from Rule base. Forward-chaining inference is often called data driven.

‡ The algorithm proceeds from a given situation to a desired goal, adding new assertions (facts) found.

‡ A forward-chaining, system compares data in the working memory against the conditions in the IF parts of the rules and determines which rule to fire.

‡ It is also known as Data Driven.



‡ Example : Forward Channing

- Given : A Rule base contains following Rule set

Rule 1:	If A and C	Then	F
Rule 2:	If A and E	Then	G
Rule 3:	If B	Then	E
Rule 4:	If G	Then	D

- Problem : Prove

  If A and B true Then D is true

- **Solution** :

  (i) ‡ Start with input given **A, B** is true and then start at

  ‡ **Rule 1** and go forward/down till a rule "fires" is found.

  First iteration :

  (ii) ‡ **Rule 3** fires : conclusion **E** is true

  ‡ new knowledge found

  (iii) ‡ No other rule fires;

  ‡ end of first iteration.

  (iv) ‡ Goal not found;

  ‡ new knowledge found at (ii);

‡ go for second iteration

Second iteration:

(v) ‡ **Rule 2** fires : conclusion **G** is true

‡ new knowledge found

(vi) ‡ **Rule 4** fires : conclusion **D** is true

‡ Goal found;

‡ Proved

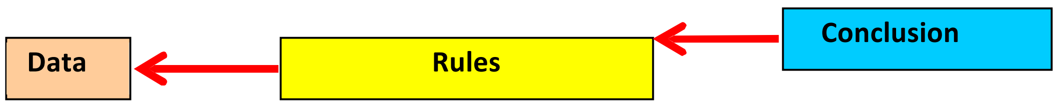
BackwardChainingAlgorithm

Backward chaining is a techniques for drawing inferences from Rule base. Backward-chaining inference is often called goal driven.

‡ The algorithm proceeds from desired goal, adding new assertions found.

‡ A backward-chaining, system looks for the action in the THEN clause of the rules that matches the specified goal.

‡ It is also known as Goal Driven



a = 1	**if a = 1 & b = 2**	**then c = 3,**	
b = 2	**if c = 3**	**then d = 4**	**d = 4**

‡ Example : Backward Channing

- Given : Rule base contains following Rule set

Rule 1:	If A and C	Then	F
Rule 2:	If A and E	Then	G
Rule 3:	If B	Then	E
Rule 4:	If G	Then	D

- Problem : Prove

If A and B true Then D is true

■ **Solution** :

(i) ‡ Start with goal ie **D** is true

‡ go backward/up till a rule "fires" isfound.

First iteration :

(ii) ‡ **Rule 4** fires :

‡ new sub goal to prove **G** is true

‡ go backward

(iii) ‡ **Rule 2** "fires"; conclusion: **A** is true

‡ new sub goal to prove **E** is true

‡ go backward;

(iv) ‡ no other rule fires; end of first iteration.

‡ new sub goal found at (iii);

‡ go for second iteration

Second iteration:

(v) ‡ **Rule 3** fires :

‡ conclusion **B** is true (2nd input found)

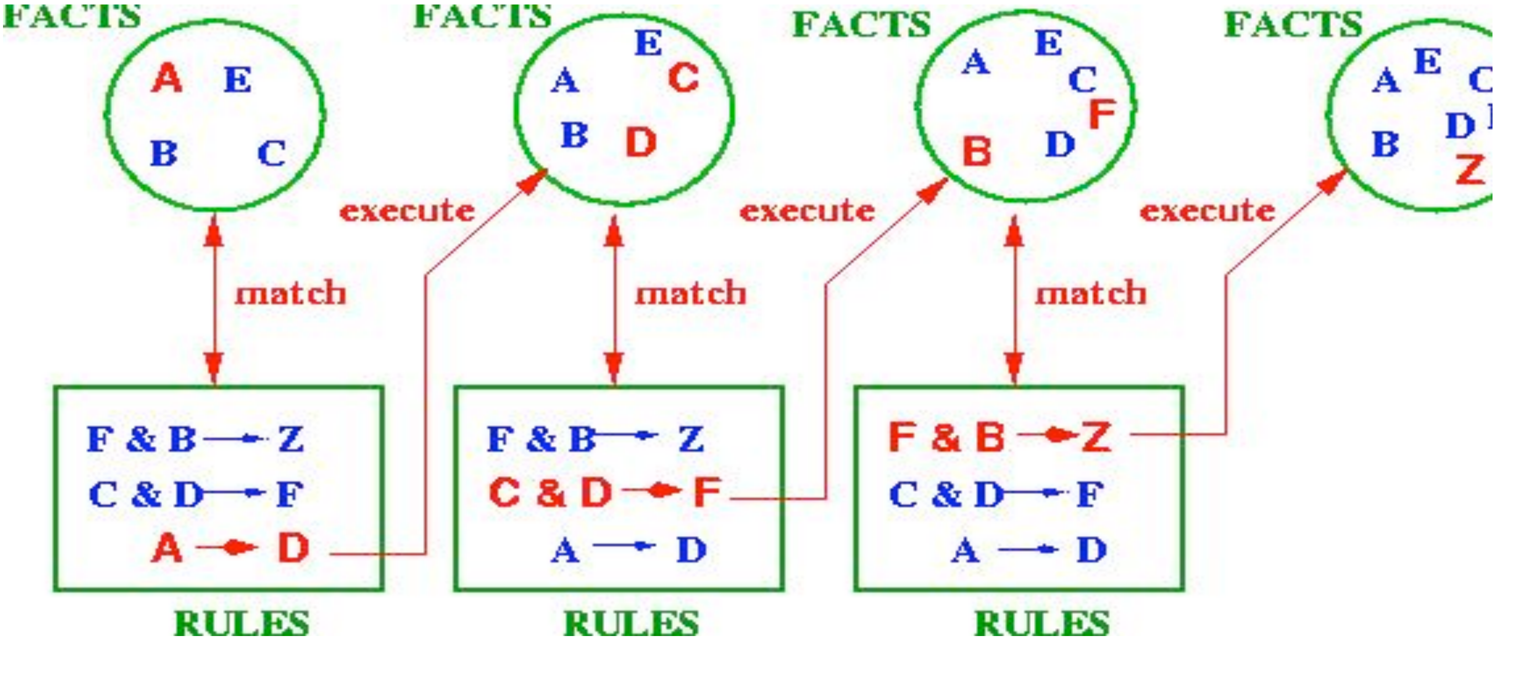
‡ both inputs **A** and **B** ascertained

‡ Proved

Forward Chaining

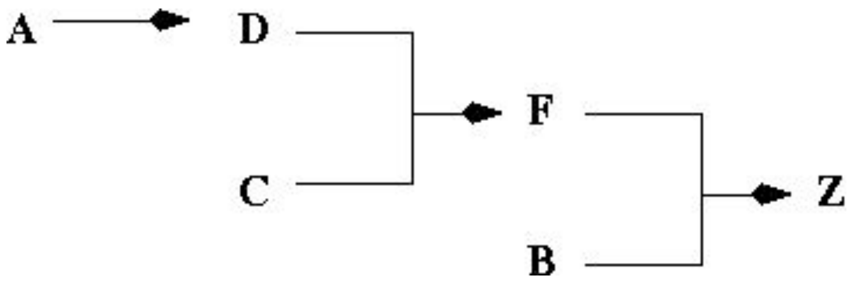
Forward chaining or **data-driven** inference works from an initial state, and by looking at the premises of the rules (IF-part), perform the actions (THEN-part), possibly updating the knowledge base or working memory.

This continues until no more rules can be applied or some cycle limit is met, e.g.



Forward Chaining (Cont'd)

In example: no more rules, that is, inference chain for this is:



Problem with forward chaining:
many rules may be applicable. The whole process is **not directed** towards a **goal**.

Backward Chaining

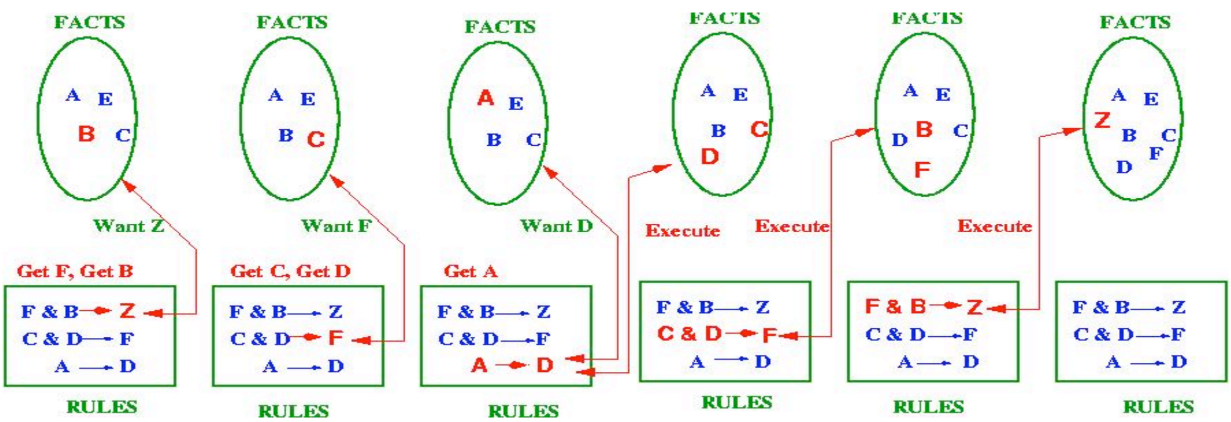
Backward chaining or **goal-driven** inference works towards a final state, and by looking at the working memory to see if goal already there. If not look at the actions (THEN-parts) of rules that will establish goal, and set up subgoals for achieving premises of the rules (IF-part).

This continues until some rule can be applied, apply to achieve goal state.

Advantage of backward chaining:
search is directed
Disadvantage of backward chaining:
goal has to be known

Now look at the example from above with backward chaining

Backward Chaining (Cont'd)



Statistical Reasoning

- An important goal for many problem-solving systems is to collect evidence as the system goes along and to modify its behavior on the basis of the evidence.
- To model this, a statistical theory of evidence is needed.

- Bayesian statistics is such a theory. The fundamental notion of Bayesian statistics is that of conditional probability
  - P(H\E)
- To compute this, the prior probability of H and the extent to which E provides evidence of H is needed.
  - P(Hi\E)- the probability that hypothesis H_i is true given evidence E.
  - P(E\H_i)- the probability that the evidence E is observed given that hypothesis is true.
  - P(H_i)- the a priori probability that hypothesis i is true in the absence of any specific evidence. These probabilities are called prior probabilities.
  - k- the number of possible hypothesis.
    Bayes' theorem then states that

$$P(H_i \backslash E) = \frac{P(E \mid H_i) \cdot P(H_i)}{\sum_{n=1}^{k} P(E \mid H_n) \cdot P(H_n)}$$

  -

Suppose for example, for solving medical diagnosis problem, consider the following assertions:

- S: patient has spots
- M: patient has measles
- F: patient has high fever

✓ Without any additional evidence, the presence of spots serves as evidence in favor of measles.
✓ Spot is also serves as a evidence for fever since measles would also cause fever.
✓ Suppose that the patient already has measles, then the additional evidence that he has spots actually tells us about the likelihood of fever.
✓ Alternatively either spots alone or fever alone cause evidence in favor of
✓ measles.
✓ If both are present then we need to find total weight of evidence.
✓ But, since spots and fever are not independent events, we need to represent conditional probability that arises from their conjunction.
✓ Foreg. Given a prior body of evidence e and some new observation E, we need to compute

$$P(H \backslash E, e) = P(H \mid E) \cdot \frac{P(e \mid E, H)}{P(e \mid E)}$$

-

The size of the set of joint probabilities grows as 2^nif there are n different propositions being considered. This makes using Bayes theorem intractable for several reasons.

- theknowledge acquisition is insurmountable.
- The space that would be required to store all the probabilities is too large
- The time required to compute the probabilities is too large.

Three things are considered:

- Attaching certainty factors to rules.
- Bayesian Networks

- Dempster-shafter theory
  - **Probability :** The Probabilities are numeric values between **0** and **1** (both inclusive) that represent ideal uncertainties (not beliefs).
    - **Probability of event A is P(A)**

$$P(A) = \frac{\text{instances of the event A}}{\text{total instances}}$$

P(A) = 0 indicates total uncertainty in **A**,

P(A) = 1 indicates total certainty and

0< P(A) < 1 values in between tells degree of uncertainty

Example 1 : A single 6-sided die is rolled.

What is the probability of each outcome?

What is the probability of rolling an even number? What is the probability of rolling an odd number?

The possible outcomes of this experiment are 1, 2, 3, 4, 5, 6. The Probabilities are:

P(1) =No of ways to roll 1 / total no of sides = 1/6

P(2) =No of ways to roll 2 / total no of sides = 1/6

P(3) =No of ways to roll 3 / total no of sides = 1/6

P(4) =No of ways to roll 4 / total no of sides = 1/6

P(5)=No of ways to roll 5 / total no of sides = 1/6

P(6) =No of ways to roll 6 / total no of sides = 1/6

P(even) =ways to roll even no / total no of sides = 3/6 = 1/2

P(odd) =ways to roll odd no / total no of sides = 3/6 =1/2

■ **Conditional probability P(A|B)**

A conditional probability is the probability of an event given that another event has occurred.

Example : Roll two dices.

What is the probability that the total of two dice will be greater than 8 given that the first die is a 6 ?

First List of the **joint possibilities** for the two dices are:

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	**(6, 3)**	**(6, 4)**	**(6, 5)**	**(6, 6)**

There are 6 outcomes for which the first die is a 6, and of these, there are 4 outcomes that total more than 8 are (6,3; 6,4; 6,5; 6,6).

The probability of a total > 8 given that first die is 6 is therefore 4/6 = 2/3 .

This probability is written as: $\underbrace{P(\text{total}>8}_{\textbf{event}} \,|\, \underbrace{\text{1st die} = 6)}_{\textbf{condition}} = 2/3$

Read as "The probability that the total is > 8 given that die one is 6 is 2/3."

■ **Probability of A and B is P(A and B)**

The probability that events **A** and **B** both occur.

Written as **P(A|B)** , is the probability of event A given that the event B has occurred.

Two events are independent if the occurrence of one is unrelated to the probability of the occurrence of the other.

‡ **If A and B are independent**

then probability that events **A** and **B** both occur is:

P(A and B) = P(A) x P(B)

ie product of probability of **A** and probability of **B**.

‡ **If A and B are not independent**

then probability that events **A** and **B** both occur is:

P(A and B) = P(A) x P(B|A) where

P(B|A) is **conditional probability** of **B** given **A**

Example 1: P(A and B) if events A and B are independent

- Draw a card from a deck , then replace it, draw another card.
- Find probability that 1st card is Ace of clubs (event A) and 2nd card is any Club (event B).
- Since there is only one Ace of Clubs, therefore probability P(A) = 1/52.
- Since there are 13 Clubs, the probability P(B) = 13/52 = 1/4.

§ Therefore, **P(A and B) =** p(A) x p(B) = 1/52 x 1/4 = 1/208.

Example 2: P(A and B) if events A and B are not independent

- Draw a card from a deck, not replacing it, draw another card.
- Find probability that both cards are Aces ie the 1st card is Ace (event A) and the 2nd card is also Ace (event B).
- Since 4 of 52 cards are Aces, therefore probability P(A) = 4/52 = 1/13.
- Of the 51 remaining cards, 3 are aces. so, probability of 2nd card is also Ace (event B) is P(B|A) = 3/51 = 1/17.

§ Therefore, **P(A and B)** = p(A) x p(B|A) = 1/13 x 1/17 = 1/221

Certainty factors and Rule-based systems

Certainty Factor: It is a measure of the extent to which the evidence that is described by the antecedent of the rule supports the conclusion that is given in the rule's consequent.

MYCIN is an example of an expert systems since it performs a task normally done by a human expert. Probabilistic reasoning is done in MYCIN. MYCIN represents most of its diagnostic knowledge as a set of rules. Each rule has associated with it a certainty factor.

Typical MYCIN rule looks like:

```
If:   (1) the stain of the organism is gram-positive, and
      (2) the morphology of the organism is coccus, and
      (3) the growth conformation of the organism is clumps,
          then there is suggestive evidence (0.7) that
          the identity of the organism is staphylococcus.
```

This is the form in which rules are stated to the user. They are actually represented internally in an easy-to-manipulate LISP list structure. The rule is represented internally as

```
PREMISE:  ($AND   (SAME CNTXT GRAM GRAMPOS)
                  (SAME CNTXT MORPH COCCUS)
                  (SAME CNTXT CONFORM CLUMPS))
ACTION:           (CONCLUDE CNTXT IDENT STAPHYLOCOCCUS TALLY 0.7)
```

MYCIN uses backward reasoning to find the disease causing organism. Once it finds the organism, it then attempts to select a therapy by which the disease may be treated. once it finds the identities of such organisms, it then attempts to select a therapy by which disease(s) may be treated.

To understand how MYCIN exploits uncertain information, we need to answer:

- What do certainty factors mean?
- How does MYCIN combine the estimates of certainty in each of its rules to produce a final estimate of the certainty of its conclusions?
  A further question that is needed to be answered is "What compromises does the MYCIN technique make and what risks are associated with those compromises?

A certainty factor (CF[h,e]) is defined in terms of two components:

- $MB[h, e]$—a measure (between 0 and 1) of belief in hypothesis h given the evidence e. MB measures the extent to which the evidence supports the hypothesis. It is zero if the evidence fails to support the hypothesis.
- $MD[h,e]$—a measure (between 0 and 1) of disbelief in hypothesis h given the evidence e. MD measures the extent to which the evidence supports the negation of the hypothesis. It is zero if the evidence supports the hypothesis.

From these two measures, we can define the CF as

$$CF[h, e] = MB[h, e] - MD[h, e]$$

- Any particular piece of evidence either supports or denies a hypothesis, and since each MYCIN rule corresponds to one piece of evidence, a single number suffices each rule to define both the MB and MD and thus the CF.
- As MYCIN reasons, CF's need to be combined to reflect the operation of multiple pieces of evidence and multiple rules applied to a problem.
- Fig. illustrates three combination scenarios that we need to consider.

(a) (b) (c)

Fig. *Combining Uncertain Rules*

Fig illustrates three combination scenarios that we need to consider

Fig(a) – several rules all provide evidence that relatez to a singlr hypothesis

Fig(b) – considers the belief in collection of several propositions taken together.

Fig© - output of one rule provides input to another

Properties to be satisfied to combine these functions in some order are:

- Since the order in which evidence is collected is arbitrary, the combining functions shou commutative and associative.
- Until certainty is reached, additional confirming evidence should increase *MB* (and similarl disconfirming evidence and *MD*).
- If uncertain inferences are chained together, then the result should be less certain than either c inferences alone.

- Consider the scenario one shown in Fig(a) in which several pieces of evidence are combined to determine the CF of one hypothesis
  The measure of belief and disbelief of a hypothesis given two observations s1 and s2 are

  computed from:

$$MB[h, s_1 \wedge s_2] = \begin{cases} 0 & \text{if } MD[h, s_1 \wedge s_2] = 1 \\ MB[h, s_1] + MB[h, s_2] \cdot (1 - MB[h, s_1]) & \text{otherwise} \end{cases}$$

$$MD[h, s_1 \wedge s_2] = \begin{cases} 0 & \text{if } MB[h, s_1 \wedge s_2] = 1 \\ MD[h, s_1] + MD[h, s_2] \cdot (1 - MD[h, s_1]) & \text{otherwise} \end{cases}$$

the measure of belief in h is 0 if h is disbelieved with certainty. Otherwise the measure of belief in h given two observations is the measure of belief given only one observation plus some increment for the second observation. This increment is computed by first taking the difference between 1 (certainty) and the belief given only the first observation. This difference is the most that can be added by the second observation. the difference is then scaled by the belief in h given only the second observation. from MB and MD,CF can be computed. if several sources of corroborating evidence are pooled , the absolute value of CF will increase. if conflicting evidence is introduced, the absolute value of CF will decrease.

A simple example shows how these functions operate. Suppose we make an initial observation that confirms our belief in h with $MB = 0.3$. Then $MD[h,s_1] = 0$ and $CF[h, s_1] = 0.3$. Now we make a second observation, which also confirms h, with $MB[h,s_2] = 0.2$. Now:

$$\begin{aligned} MB[h,s_1 \wedge s_2] &= 0.3 + 0.2 \cdot 0.7 \\ &= 0.44 \\ MD[h,s_1 \wedge s_2] &= 0.0 \\ CF[h,s_1 \wedge s_2] &= 0.44 \end{aligned}$$

- Consider the scenario two shown in Fig(b), in which we need to find certainty factor of combination of hypothesis.
  - The combination of certainty factor can be computed from its MB and MD.
  - The formula MYCIN uses for the MB of the conjunction and the disjunction of two hypothesis are:

$$MB[h_1 \wedge h_2, e] = \min(MB[h_1, e], MB[h_2, e])$$

$$MB[h_1 \wedge h_2, e] = max(MB[h_1,e], MB[h_2,e])$$

MD can be computed analogously

- Consider the scenario two shown in Fig(c), in which rules are chained together with the result that the uncertain outcome of rule must provide the input to another.

MYCIN provides a chaining rule that is defined as follows. Let MB'[h,s] be the measure of belief in h given the validity of s. Let e be the evidence that led us to believe in s. then

MB[h,s]=MB'[h,s].max(0,CF[s,e])

Since initial CF's in MYCIN are estimates that are given by experts who write the rules, it is not really necessary to state a more precise definition of what a CF means than the one that is already given.

$$MB[h, e] = \begin{cases} 1 & \text{if } P(h) = 1 \\ \dfrac{\max[P(h \mid e), P(h)] - P(h)}{1 - P(h)} & \text{otherwise} \end{cases}$$

Similarly, the *MD* is the proportionate decrease in belief in *h* as a result of *e*:

$$MD[h, e] = \begin{cases} 1 & \text{if } P(h) = 0 \\ \dfrac{\min[P(h \mid e), P(h)] - P(h)}{-P(h)} & \end{cases}$$

it turns out that these definitions are incompatible with a Bayesian view of conditional probability. Small changes to them, make them compatible .

MB is redefined as

$$MB[h,e] = \begin{cases} 1 & \text{if } P(h) = 1 \\ \dfrac{\max[P(h|e), P(h)] - P(h)}{(1 - P(h) \cdot P(h|e)} & \text{otherwise} \end{cases}$$

The definition of MD is also changed similarly.

With these interpretations there ceases to be any fundamental conflict between MYCIN's techniques and those suggested by Bayesian statistics. But still MYCIN works why?

Each CF in a MYCIN rule represents the contribution of an individual rule to MYCIN's belief in a hypothesis. In some sense it represents a conditional probability, P(H/E). In a pure Bayesian system, P(H/E) describes the conditional probability of H given that the only relevant evidence is E. If there is other evidence, joint probabilities need to be considered. this is where MYCIN diverges from a pure Bayesian system, with the result that it is easier to write and more efficient to execute, but with the corresponding risk that its behavior will be counter initiative. It is being assumed that all rules are independent. Each of the combination scenario is vulnerable when this independence assumption is violated.

in the example there are three antecedents, with a single CF rather than three separate rules, this makes the combination rules unnecessary. the rule writer did this because the three antecedents are not independent.

consider that there are three separate rules and the CF of each was 0.6. If we apply MYCIN combination formula to the three separate rules we get

$$\begin{aligned} MB[h, s \wedge s_2] &= 0.6 + (0.6 \cdot 0.4) \\ &= 0.84 \\ MB[h, (s_1 \wedge s_2) \wedge s_3] &= 0.84 + (0.6 \cdot 0.16) \\ &= 0.936 \end{aligned}$$

This is a substantially different result than the true value, as expressed by the expert of 0.7.

Now let's consider what happens when independence assumptions are violated in the third example. Let's consider a concrete example in which

S: sprinkler was on last night
W: grass is wet
R: it rained last night

MYCIN rules that describe the predictive relationship among these three events is

```
If:  the sprinkler was on last night
then there is suggestive evidence (0.9) that
     the grass will be wet this morning
```

This rule may accurately describe the world. Now consider the second rule.

```
If: the grass is wet this morning
then there is suggestive evidence (0.8) that
    it rained last night
```

This rule makes sense when rain is the most common source of water on the grass. but if rules are applied together using MYCIN's chaining rule, we get

$$MB[W,S] = 0.8 \qquad \{\text{sprinkler suggests wet}\}$$
$$MB[R,W] = 0.8 \cdot 0.9 = 0.72 \qquad \{\text{wet suggests rains}\}$$

In other words it is believed that it rained because it is believed that sprinkler was on.

One of the major advantages of the modularity of the MYCIN rule system is that it allows us to consider individual antecedent/consequent relationships independently of others.

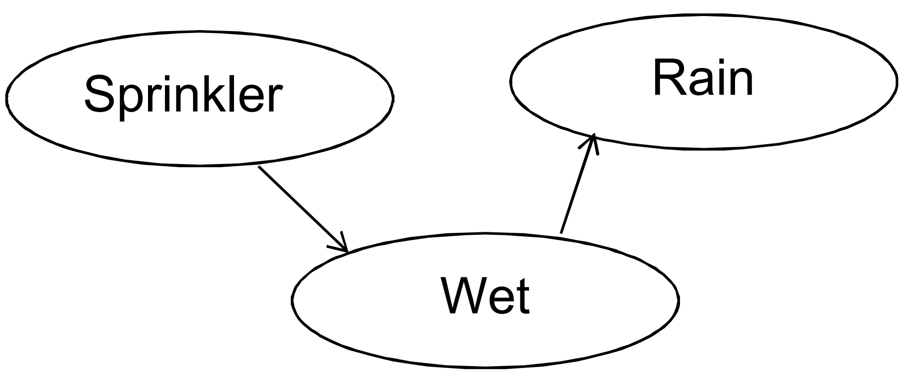
To summarize, this approach makes strong independence assumptions that make it relatively easy to use, at the same time assumptions create dangers if rules are not written carefully so that important dependencies are captured.

Bayesian Networks

It is an alternative approach to what we did in the previous section.

The idea is to describe the real world, it is not necessary to use a huge joint probability table in which we list the probabilities of all combinations, because most events are independent of each other, there is no need to consider the interactions between them. Most events are conditionally independent of most other ones, so their interactions need not be considered.

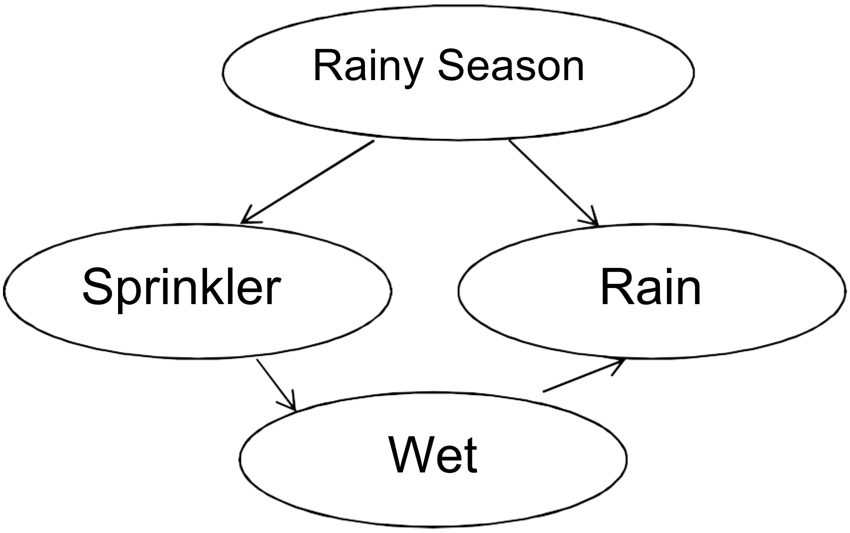
Instead a more local representation is used in which clusters of events that interact is described.



Network notation describes the various kinds of constraints on likelihoods that propositions can have on each other. The above figure shows the flow of constraints . But the problem was the direction of constraint flow.there are two different ways that propositions can influence the likelihood of each other.

- the first is that causes influence the likelihood of their symptoms.
- the second is that observing a symptom affects the likelihood of all of its possible causes.

  the idea behind the Bayesian network structure is to make a clear distinction between these two kinds of influence. Specifically we construct a directed acyclic graph that represents causality relationships among variables.



Representing causality uniformly

The above diagram is a causality graph for the wet grass example. in addition to the three nodes, the graph contains a new node corresponding to the propositional variable that tells us whether it is currently the rainy season.A DAG illustrates the causality relationships that occur among the nodes it contains. To use it as a basis for probabilistic reasoning, more information is needed. in particular it is necessary to know for each value of a parent node, what evidence is provided about the values that the child node can take on. This is stated in a table in which the conditional probabilities are provided.

Attribute	Probability
$p(Wet \mid Sprinkler, Rain)$	0.95
$P(Wet \mid Sprinkler, \neg Rain)$	0.9
$p(Wet \mid \neg Sprinkler, Rain)$	0.8
$p(Wet \mid \neg Sprinkler, \neg Rain)$	0.1
$p(Sprinkler \mid RainySeason)$	0.0
$p(Sprinkler \mid \neg RainySeason)$	1.0
$p(Rain \mid RainySeason)$	0.9
$p(Rain \mid \neg RainySeason)$	0.1
$p(RainySeason)$	0.5

Conditional probabilities for a Bayesian network

From the above table, it is seen that the prior probability of the rainy season is 0.5. Then if it is the rainy season, the probability of rain on a given night is 0.0, if it is not the probability is only 0.1. a mechanism is needed for computing the influence of any arbitrary node on any other.

Suppose it rained last night,what is the probability that it is the rainy season?to answer this question it requires that the initial DAG to be converted to an undirected graph in which the arcs can be used to transmit probabilities in either direction depending on where the evidence is coming from.also a mechanism is needed for using the graph to guarantee that the probabilities are transmitted correctly.

For example while it is true that observing wet grass may be evidence for rain and observing rain is evidence for wet grass, it must be guaranteed that no cycle is ever traversed in such a way that wet grass is evidence for rain which is then taken as evidence for wet grass and so forth.

Three algorithms are available for doing these computations

A message-passing method.

A cliché triangulation method.

A variety of stochastic algorithms.

A message-passing method is based on the observation that to compute the probability of a node A given what is known about the other nodes in the network it is necessary to know three things

 The total support arriving at A from its parents.

 The total support arriving at A from its children.

 The entry in the fixed condition probability matrix that relates A to its causes.

Cliché triangulation method. Explicit arcs are introduced between pair of nodes that share a common descendent.

Stochastic Algorithm or Randomized Algorithms – The idea is to shield a given node probabilistically from most of the other nodes in the network. These algorithms run faster but may not give correct results.

Dempster- Shafer theory

- ❖ The Dempster-Shafer theory, also known as the theory of belief functions, is a generalization of the Bayesian theory of subjective probability.
- ❖ Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities;
- ❖ The Dempster-Shafer theory owes its name to work by A. P. Dempster (1968) and Glenn Shafer (1976), but the theory came to the attention of AI researchers in the early 1980s,

when they were trying to adapt probability theory to expert systems.

- ❖ Dempster-Shafer degrees of belief resemble the certainty factors in MYCIN, and this resemblance suggested that they might combine the rigor of probability theory with the flexibility of rule-based systems.
- ❖ The Dempster-Shafer theory remains attractive because of its relative flexibility.
- ❖ DST is a mathematical **theory of evidence** based on belief functions and plausible reasoning. It is used to combine separate pieces of information (evidence) to calculate the probability of an event.
- ❖ DST offers an alternative to traditional probabilistic theory for the mathematical representation of uncertainty.
- ❖ DST can be regarded as, a more general approach to represent uncertainty than the Bayesian approach.
- ❖ Bayesian methods are sometimes inappropriate
- ❖ This new approach considers sets of propositional and assigns to each of them an interval

  [Belief,Plausibility]

  in which the degree of belief must lie. Belief (usually denoted Bel) measures the strength of the evidence in favor of a set of propositions.
- ❖ It ranges from 0(indicating no evidence) to 1(denoting evidence)
- ❖ Plausibility(Pl) is defined as
  - o Pl(s)=1-Bel(¬s)
- ❖ It also ranges from 0 to 1 and measures the extent to which evidence in favor of ¬s leaves room for belief in s.
- ❖ In particular, if there is certain evidence in favor of ¬s, then Bel(¬s) will be 1 and Pl(s) will be 0.
- ❖ **Belief** in a hypothesis is constituted by the sum of the masses of all sets enclosed by it (i.e. the sum of the masses of all subsets of the hypothesis). It is the amount of belief that directly supports a given hypothesis at least in part, forming a lowerbound.
- ❖ **Plausibility**is1minus the sum of the masses of all sets whose intersection with the hypothesis is empty. It is an upper bound on the possibility that the hypothesis could possibly happen, up to that value, because there is only so much evidence that contradicts that hypothesis.
- ❖ The Dempster-Shafer theory is based on two ideas:
  - ➢ the idea of obtaining degrees of belief for one question from subjective probabilities for a relatedquestion,
  - ➢ Dempster's rule for combining such degrees of belief when they are based on independent items ofevidence.
- ❖ Frame of discernment is defined as the exhaustive universe of mutually exclusive hypothesis and it is written as Θ.

- For example, Θ might consists of the set [All, flu, cold,Pneu]

  All: allergy
  Flu: flu
  Cold: cold
  Pneu: pneumonia

- \

- The goal is to attach some measure of belief to elements of Θ. Often it supports sets of elements (i.e subsets of Θ). For example, in the diagnosis problem, fever might support{flu,cold,pneu}.
- Dempster-shafter theory lets us handle interactions by manipulating sets of hypothesis directly.
- The key function that is used is a probability density function, which is denoted as m.
- The function m is defined not just for elements of Θ but for all subsets of it.(including singleton elements)
- The quantity m(p) measures the amount of belief that is currently assigned to exactly the set p of hypothesis.
- If Θ contains n elements, then there are 2^nsubsets of Θ. The value of m is assigned so that the sum of all the m values assigned to the subsets of Θ is 1.
- Let us see how m works for the diagnosis problem. Assume that there is no information about how to choose among the four hypothesis. Then we define m as
  [Θ] (1.0)
- All other values of m are thus 0. Suppose, a piece of evidence is acquired that suggests(at a level of 0.6) that the correct diagnosis is in the set [Flu, Cold, Pneu]. Fever might be such a piece of evidence. m is updated as follows

$$\{Flu, Cold, Pneu\} \qquad (0.6)$$
$$\{\Theta\} \qquad (0.4)$$

- Having defined m, Bel(p) can now be defined for a set p as the sum of the values of m for p and for all of its subsets. Thus Bel(p) is the overall belief.
- Suppose there are two belief functions m_1 and m_2. Let X be the set of subsets of Θ to which m_1 assigns a nonzero value and let Y be the corresponding set for m_2.
- The combination m_3 of m_1 and m_2 is given as

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \phi} m_1(X) \cdot m_2(Y)}$$

-
- For example suppose m_1 corresponds to our belief after observing fever

$\{Flu, Cold, Pneu\}$	(0.6)
Θ	(0.4)

- suppose m_2 corresponds to our belief after observing runny nose

$\{All, Flu, Cold\}$	(0.8)
Θ	(0.2)

- then $m_3$is computed using the following table which is derived using the numerator of the combination rule.

		$\{A, F, C\}$	(0.8)	Θ	(0.2)
$\{F, C, P\}$	(0.6)	$\{F, C\}$	(0.48)	$\{F, C, P\}$	(0.12)
Θ	(0.4)	$\{A, F, C\}$	(0.32)	Θ	(0.08)

$\{Flu, Cold\}$	(0.48)
$\{All, Flu, Cold\}$	(0.32)
$\{Flu, Cold, Pneu\}$	(0.12)
Θ	(0.08)

Let $m_4$correspond to the belief given just the evidence that the problem goes away when the patient goes on a trip:

$\{All\}$	(0.9)
Θ	(0.1)

		{A}	(0.9)	Θ	(0.1)
{F, C}	(0.48)	Φ	(0.432)	{F, C}	(0.048)
{A, F, C}	(0.32)	{A, F, C}	(0.288)	{A, F, C}	(0.032)
{F, C, P}	(0.12)	Φ	(0.108)	{F, C, P}	(0.012)
Θ	(0.08)	{A}	(0.072)	Θ	(0.008)

There is now a total belief of 0.54 associated with Φ. So it is necessary to scale the remaining values by the factor 1-0.54=0.46. if this is done and also combine alternative ways of generating the set [All, flu,cold], then the final combined belief function m_5 is given by

{Flu, Cold}	(0.104)
{All, Flu, Cold}	(0.696)
{Flu,Cold,Pneu}	(0.026)
{All}	(0.157)
Θ	(0.017)

Fuzzy Reasoning

What is Fuzzy Set?

- The word "fuzzy" means "vagueness". Fuzziness occurs when the boundary of a piece of information is notclear-cut.
- Fuzzy sets have been introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set.
- Classical set theory allows the membership of the elements in the set in binary terms, a bivalent condition - an element either belongs or does not belong to the set.

  Fuzzy set theory permits the gradual assessment of the membership of elements in a set, described with the aid of a membership function valued in the real unit interval [0, 1].
- **Example:**

  Words like young, tall, good, or high are fuzzy.

  - There is no single quantitative value which defines the term young.

- For some people, age 25 is young, and for others, age 35 is young.
- The concept young has no clean boundary.
- Age1isdefinitelyyoungandage100isdefinitelynotyoung;
- Age35 has some possibility of being young and usually depends on the context in which it is being considered.

Introduction

In real world ,there exists much fuzzy knowledge;

Knowledge that is vague, imprecise, uncertain, ambiguous, inexact, or probabilistic in nature.

Human thinking and reasoning frequently involve fuzzy information, originating from inherently inexact human concepts. Humans, can give satisfactory answers, which are probably true.

However, our systems are unable to answer many questions. The reason is, most systems are designed based upon classical set theory and two-valued logic which is unable to cope with unreliable and incomplete information and give expert opinions.

- **Classical Set Theory**

A Set is any well defined collection of objects. An object in a set is called an element or member of that set.

- Sets are defined by a simple statement describing whether a particular element having a certain property belongs to that particular set.
- Classical set theory enumerates all its elements
- using

$$\mathbf{A = \{ a1 , a2 , a3 , a4 , ... an \}}$$

If the elements **ai (i = 1, 2, 3, . . . n)** of a set **A** are subset of universal set **X**, then set **A** can be represented for all elements **x ∈ X** by its **characteristic function**

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases}$$

- A set **A** is well described by a function called characteristic function.

  This function, defined on the universal space **X**, assumes:

  a value of **1** for those elements **x** that belong to set **A**, and

  a value of **0** for those elements **x** that do not belong to set **A**.

  The notation is used to express these mathematically are

$$\left.\begin{array}{l} \mathbf{A : X \rightarrow [0, 1]} \\ \mathbf{A(x) = 1 ,\ x \text{ is a member of } A} \\ \mathbf{A(x) = 0 ,\ x \text{ is not a member of } A} \end{array}\right\} \quad \textbf{Eq.(1)}$$

  Alternatively, the set **A** can be represented for all elements $x \in X$ by its characteristic function $\mu_A(x)$ defined as

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{Eq.(2)}$$

- Thus in classical set theory $\mu_A(x)$ has only the values **0** ('false') and **1** ('true''). Such sets are called **crisp sets.**

- **Fuzzy Set Theory**

  Fuzzy set theory is an extension of classical set theory where elements have varying degrees of membership. A logic based on the two truth values, *True* and *False*, is sometimes inadequate when describing human reasoning. Fuzzy logic uses the whole interval between **0**(false) and1 (true)to describe human reasoning.

  - A **Fuzzy Set** is any set that allows its members to have different degree of membership, called membership **function**, *in* the interval **[0 ,1]**.
  - The **degree of membership** or truth is not same as probability;
    - fuzzy truth is not likelihood of some event or condition.
    - fuzzy truth represents membership in vaguely defined sets;
  - **Fuzzy logic** is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicatelogic.
  - Fuzzy logic is capable of handling inherently imprecise concepts.
  - Fuzzy logic allows in linguistic form the set membership values to imprecise concepts like**"slightly","quite"and"very".**
  - Fuzzy set theory defines Fuzzy Operators on Fuzzy Sets.

- **Examples of Crisp**

  Example 1: Set of prime numbers (a crisp set)

  If we consider space **X** consisting of natural numbers **≤ 12**

  ie **X = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}**

  Then, the set of prime numbers could be described as follows.

  PRIME = {x contained in X | x is a prime number} = {2, 3, 5, 6, 7, 11}

Fuzzy set Terminology

Universe of Discourse

This is defined as the range of all possible values that comprise the input to the fuzzy system.

Fuzzy Set

A **Fuzzy Set** is any set that allows its members to have different degree of membership, called **membership function**, in the interval **[0 ,1]**.

- **Support of a fuzzy set (S_f):** The support S of a fuzzy set f in a universal crisp set U is that set which contains all elements of the set U that have a non-zero membership value in f

the support of the fuzzy set adult S_{adult} is given by

$S_{adult} = \{21,30,35,40,45,60,70\}$

Depiction of a fuzzy set: A fuzzy set in a universal crisp set U is written as

$f = \mu 1/s1 + \mu 2/s2 + \ldots + \mu n/sn$ wher μ_i is the membership, s_i is the corresponding term in the support set ; + and / are only user for representation purpose; fuzzy set OLD is depicted as

Old =0.1/21+0.3/30+0.35/35+0.4/40+0.6/45+0.8/60+1/70

Fuzzy Operations

A fuzzy set operations are the operations on fuzzy sets. The fuzzy set operations are generalization of crisp set operations. Zadeh [1965] formulated the fuzzy set theory in the terms of standard operations: Complement, Union, Intersection, and Difference.

- Union: The membership function of the union of two fuzzy sets A and B is defined as the maximum of the two individual membership functions. It is equivalent to the Boolean OR operation $\mu_{AUB} = \max(\mu_A, \mu_B)$
- Intersection: The membership function of the Intersection of two fuzzy sets A and B is defined as the minimum of the two individual membership functions. It is equivalent to the Boolean AND operation

$$\mu_{A^\wedge B} = \min(\mu_A, \mu_B)$$

- Complement: The membership function of the complement of a fuzzy set A is defined as the negation of the specified membership function It is equivalent to the Boolean NOT operation $\mu\ A^c = (1 - \mu_A)$

Fuzzy Inference Processing

- There are three models for Fuzzy processing based on the expressions of consequent parts in fuzzy rules

Suppose x_i are inputs and y is the consequents in fuzzy rules

1. Mamdani Model: y = A

where A is a fuzzy number to reflect fuzziness

- Though it can be used in all types of systems, the model is more suitable for knowledge processing systems than control systems

2. TSK (Takagi-Sugano-Kang) model:

$y = a0 + \Sigma\, ai\, xi$ where ai are constants

- The output is the weighted linear combination of input variables (it can be expanded to nonlinear combination of input variables)
- Used in fuzzy control applications

3. Simplified fuzzy model: y = c

where c is a constant

- Thus consequents are expressed by constant values

Applications of Fuzzy Logic

- Fuzzy logic has been used in many applications including

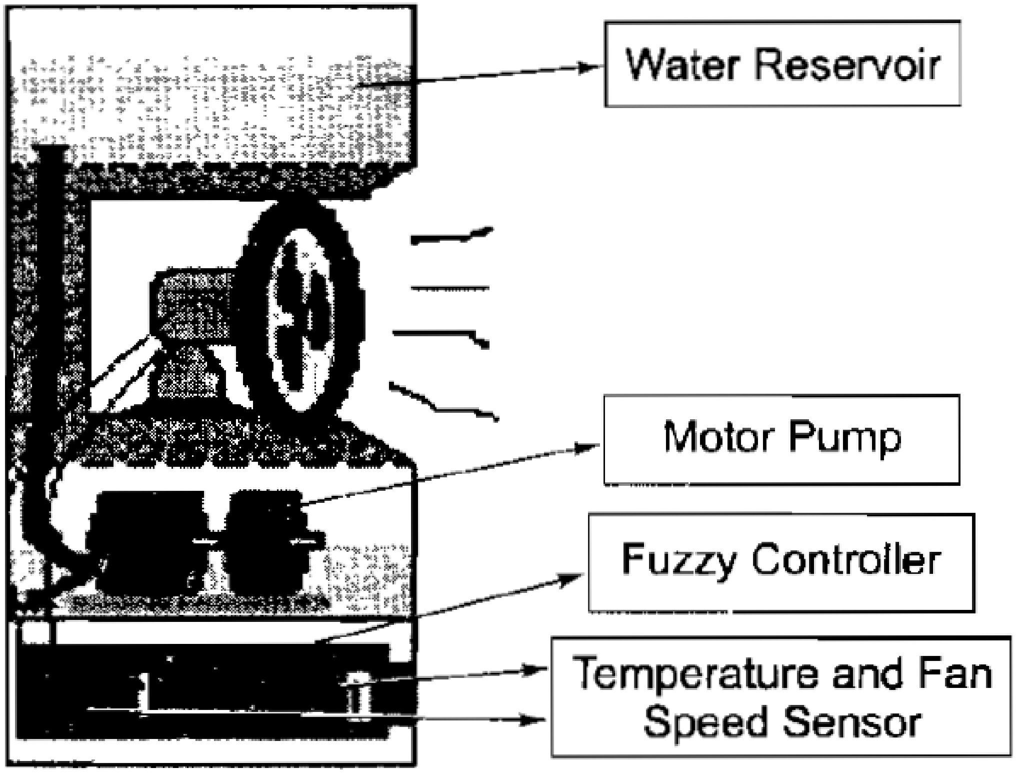
  - Domestic appliances like washing machines and cameras

  -Sophisticated applications such as turbine control, data classifiers etc.

  - Intelligent systems that use fuzzy logic employ techniques for learning and adaptation to the environment.

Case Study: Controlling the speed of a motor in a room cooler

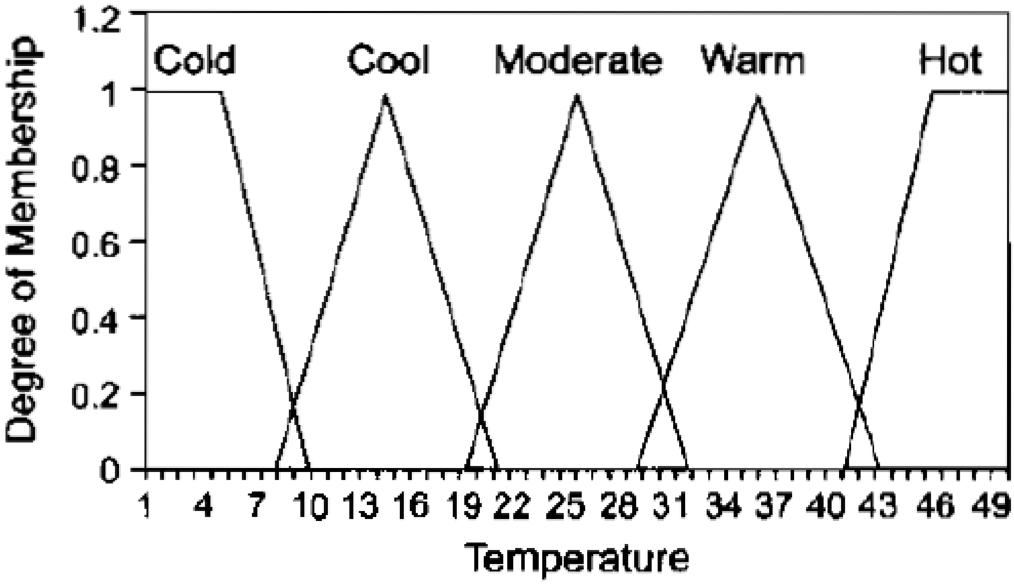
- In this case study,the speed of the motor is based on two parameters: temperature and humidity; humidity is increased to reduce temperature.Mamdani style of inference processing is used.this uses a tank of water and a fan to increase humidity to bring down temperature.
- This case study helps to understand fuzzy logic, defining fuzzy rules and fuzzy inference and control mechanisms
- **Fuzzy Room Cooler**
- A room cooler has a fan encased in a box with wool or hay. The wool is continuously moistened by water that flows through a pump connected to a motor.

- A motorized pump controls the rate of flow of water required for moistening.
- The rate of flow of water is to be determined; it is a function of room temperature and the speed of motor
- Two sensors mounted inside the cooler or in the room at strategic locations measure the fan motor speed and the temperature within the room.
- The fan speed could be varied either by a knob by the user or could be designed to change based on an appropriate parameter sensed , humidity for instance.
- 

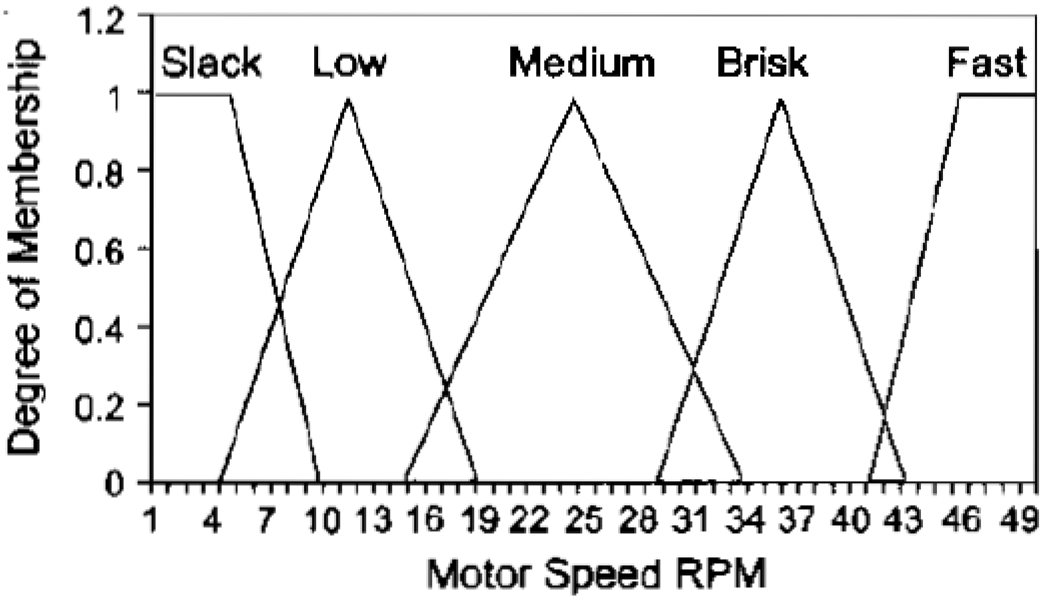
Fuzzy Regions

- Two parameters viz temperature and pressure decide the water flow rate.
- Fuzzy terms for temperature is given as cold, cool, moderate,warm and hot
- Fuzzy terms for fan speed (measured in rotations per minute) as slack,low,medium,brisk,fast.
- The output of the system, which is the flow rate of the water controlled by the motorized pump could also be defined accordingly by another set of fuzzy terms- Strong Negative (SN), Negative (N),Low-Negative (LN), Medium (M), Low-Positive(LP), Positive (P), and High-Positive (HP)
- **Fuzzy Profiles**

- Fuzzy profiles are defined for each of the three parameters by assigning memberships to their respective values
- The profiles have to be carefully designed after studying the nature and desired behavior of the system.



- In the above diagram, when the temperature is 25 degree its membership to the fuzzy set **moderate** is 1. But as it gets drifted to 30 degrees, its membership to this set decreases while the same to the set **warm** sets to increase.
- Thus when the temperature is 30 degrees it is neither fully moderate not warm.



- *(b) Fan Motor Speed*

Fuzzy relationships for the inputs Fan Motor speed

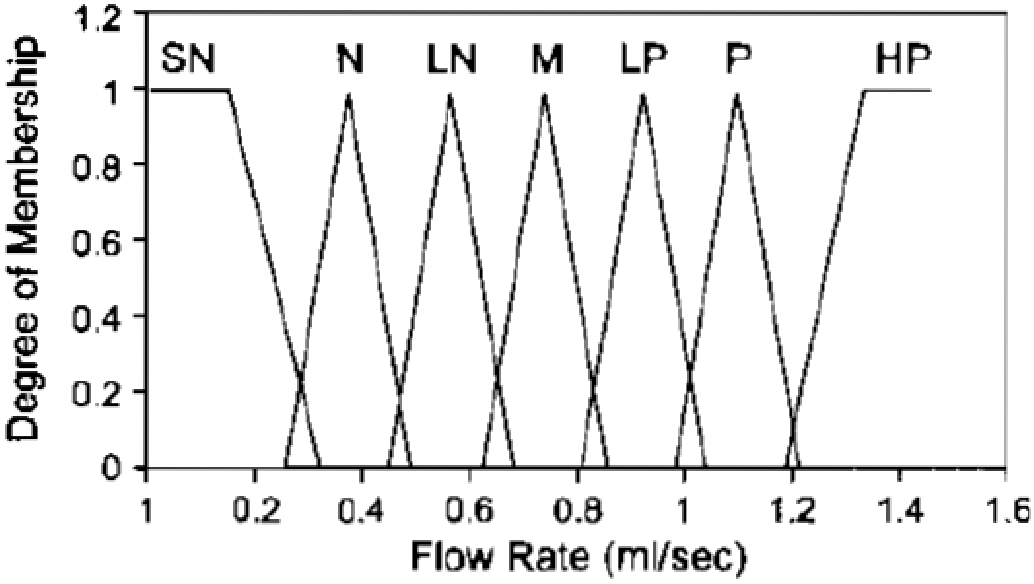


Fig. 22.3 *Water Flow Rate*

-

Fuzzy relationships for the outputs Water Flow Rate

Fuzzy Rules

- The fuzzy rules form the triggers of the fuzzy engine
- After a study of the system, the rules could be written as follows
- R1: If temperature is HOT and fan motor speed is SLACK then the flow-rate is HIGH-POSITIVE
- R2: If temperature is HOT and fan motor speed is LOW then the flow-rate is HIGH-POSITIVE
- R3: If temperature is HOT and fan motor speed is MEDIUM then the flow-rate is POSITIVE
- R4: If temperature is HOT and fan motor speed is BRISK then the flow-rate is HIGH-POSITIVE
- R5: If temperature is WARM and fan motor speed is MEDIUM then the flow-rate is LOW-POSITIVE
- R6: If temperature is WARM and fan motor speed is BRISK then the flow-rate is POSITIVE
- R7: If temperature is COOL and fan motor speed is LOW then the flow-rate is NEGATIVE
- R8: If temperature is MODERATE and fan motor speed is LOW then the flow-rate is MEDIUM

Fuzzification

- The fuzzifier forms the heart of the fuzzy region. Whenever the sensors report the values of temperature and fan speed, they are mapped based on their memberships to the respective fuzzy regions they belong to.

- From Figure 1., the temperature 42 degrees correspond to two membership values 0.142 and 0.2 that belong to WARM and HOT fuzzy regions respectively

- Similarly From Figure 2., the fan speed 31 rpm corresponds to two membership values 0.25 and 0.286 that belong to MEDIUM and BRISK fuzzy regions respectively

Parameter	*Fuzzy Regions*	*Memberships*
Temperature Fan speed	warm, hot medium, brisk	0.142, 0.2 0.25,0.286

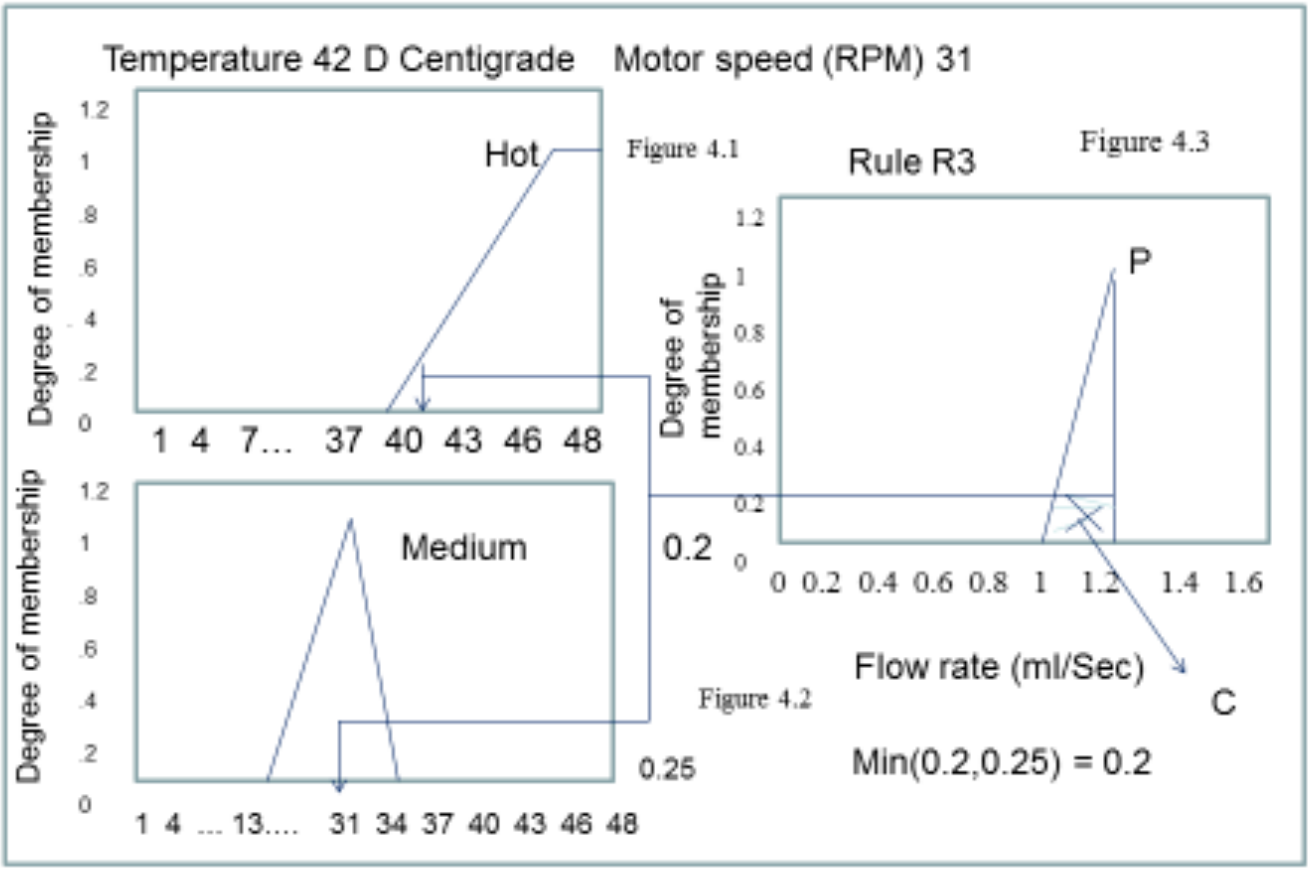
From Table 2, there are four combinations possible

- If temperature is WARM and fan speed is MEDIUM
- If temperature is WARM and fan speed is BRISK
- If temperature is HOT and fan speed is MEDIUM
- If temperature is HOT and fan speed is BRISK
- Comparing the above combinations with the left side of fuzzy rules R5, R6, R3, and R4 respectively, the flow-rate should be LOW-POSITIVE, POSITIVE, POSITIVE and HIGH-POSITIVE.
- The conflict should be resolved and the fuzzy region is to be given as a value for the parameter water flow-rate
- **Defuzzification**
- The fuzzy outputs LOW-POSITIVE, POSITIVE, and HIGH-POSITIVE are to be converted to a single crisp value that is provided to the fuzzy cooler system; this process is called defuzzification.
- Several methods are used for defuzzification
- The most common methods are
    1. The centre of gravity method and
    2. The Composite Maxima method

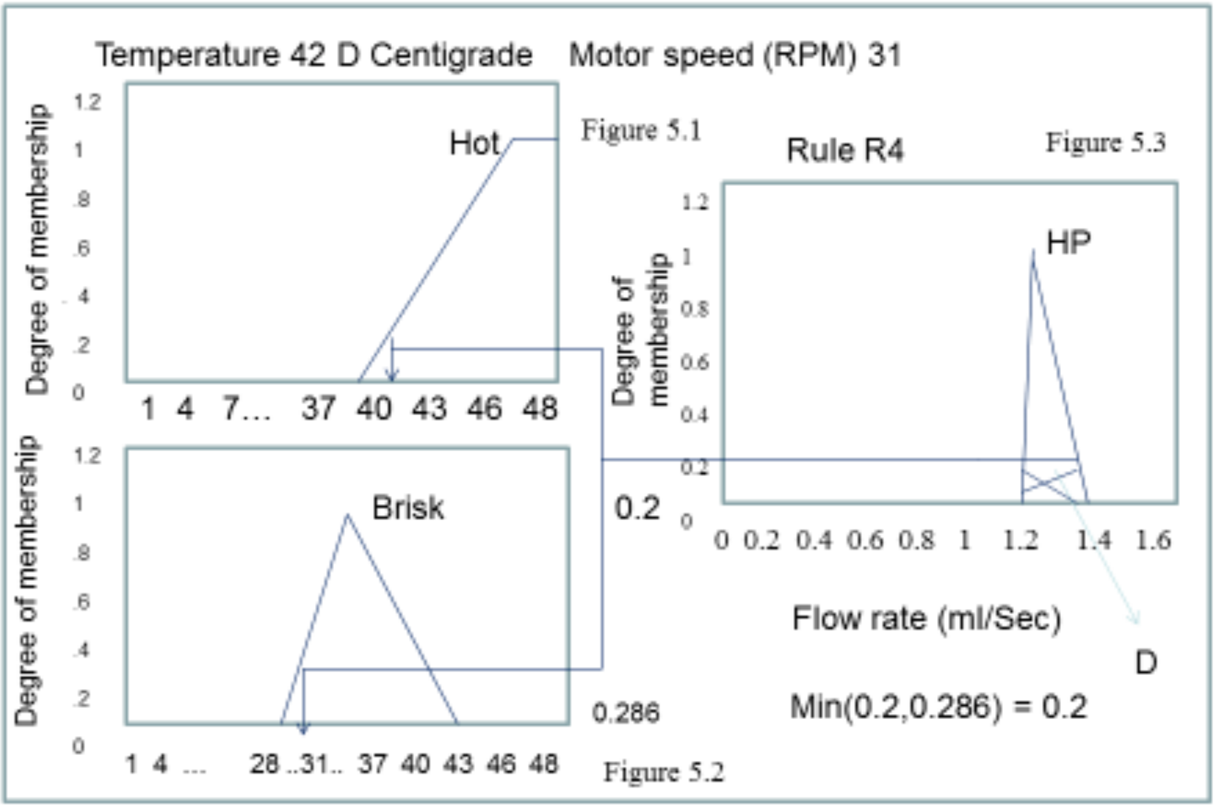
The **centroid**, of a two-dimensional shape *X* is the intersection of all straight lines that divide *X* into two parts of equal moment about the line or the average of all points of X. (Moment is a quantitative measure of the shape of a set of points.)

In both these methods the composite region formed by the portions A, B, C, and D (corresponding to rules R3, R4, R5 and R6 respectively) on the output profile is to be computed

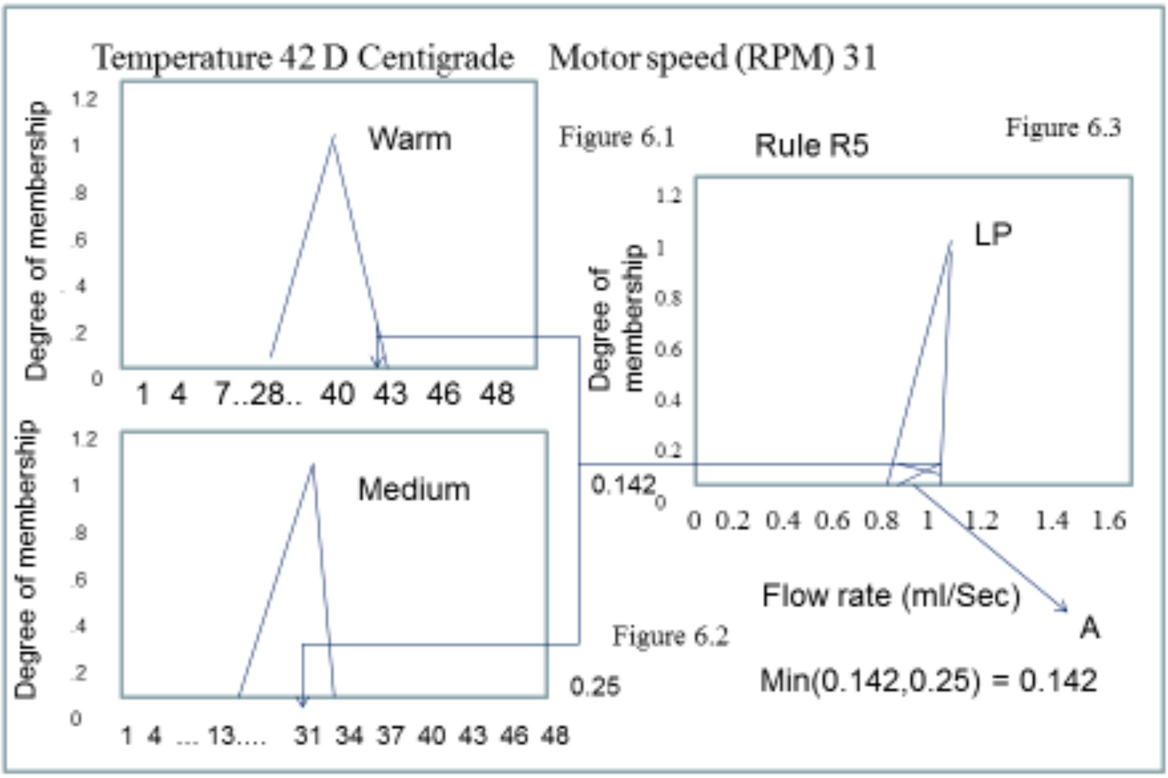
Defuzzification contd..



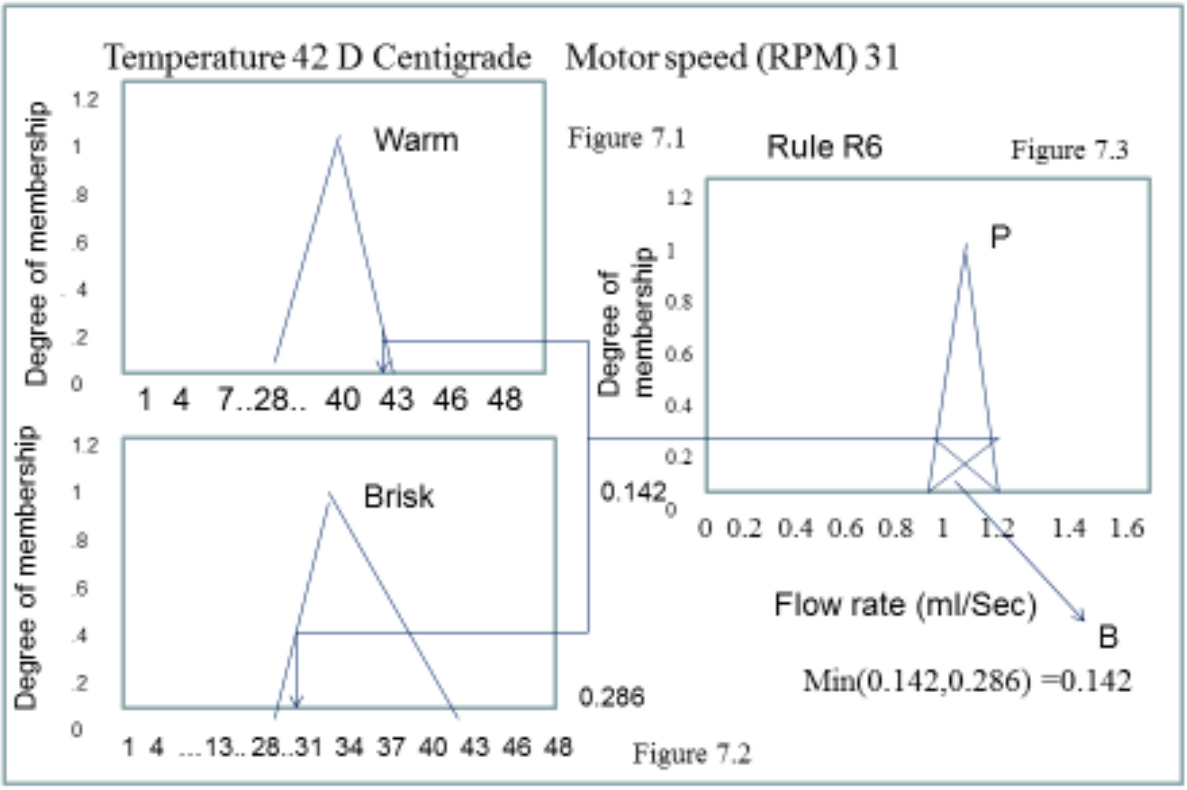
Defuzzification contd..

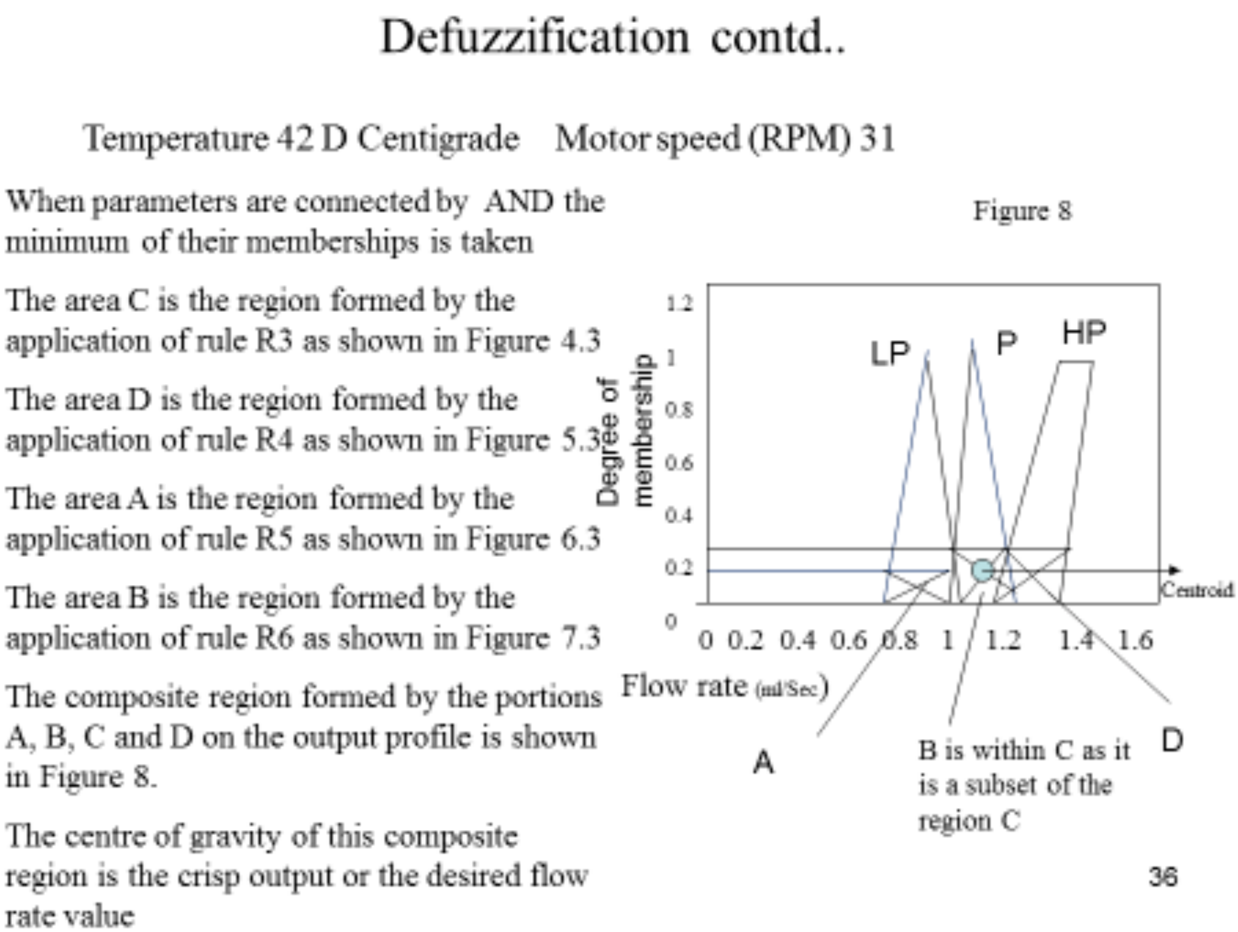


Defuzzification contd..



Defuzzification contd..





Steps in Fuzzy logic based system

- Formulating fuzzy regions
- Fuzzy rules
- Embedding a Defuzzification procedure

In Defuzzification procedure, depending on the application, either the centre of gravity or the composite maxima is found to obtain the crisp output.