B.TECH. TCS-603

SIXTH SEMESTER EXAMINATION, 2008-2009

ARTIFICIAL INTELLIGENCE

Time : 3 Hours *Total Marks :* **100**

Note : *Attempt all questions.*

1. Attempt any four parts of the following :

(a) Explain Intelligence and Artificial Intelligence. How do you distinguish between the two? Give different definition based on various forms of Intelligence.

Ans. Intelligence : Intelligence is the ability to reason, to trigger new thoughts, where thoughts can be defined as a mechanism which stimulates action, further thoughts, information generation as a mechanism which acts through present environment or past memory.

1. **Artificial Intelligence :** "AI is developing computer programs to solve complex problems by applications of processes that are analogous to human reasoning processes".

Intelligence and AI can be distinguished as :

Artificial Intelligence

1. AI is branch of computer science concerned with study and creation of computer systems that exhibit some form of intelligence.
2. It corresponds to creation of system that learn new concepts and tasks, that can reasons and draw useful conclusions, that can understand natural language etc.
3. Programmed by human beings.
4. Nothing called hereditary but systems do learn from experience.

Intelligence

1. It is the ability to reason, to trigger new thoughts, where thoughts can be defined as mechanism which stimulates action.
2. Intelligence is not involved in the creation of any kind of system. It is the inherit property of brain of human beings and other living beings, by the way they analyse the world.
3. Natural.
4. Increases with experience and also hereditary.

Ordinary tasks :

- Perception
  - vision
  - speech
- Natural language
  - understanding
  - generation
  - translation
- Common sense reasoning
- Robot control

Formal tasks :

- Games
  - Chess
  - Tic-Tac-Toe

- Mathematics
    - geometry
    - integral calculus

Expert tasks :

- Engineering
    - design
    - fault finding
    - manufacturing planning
- scientific analysis
- medical diagnosis
- financial analysis

2. **Different definitions based on various forms of intelligence :**

    Artificial Intelligence : AI is a branch of computer system in which we study about those computers that processes the leaving, thinking and decision making capabilities.

(b) Distinguish between Heuristics and Algorithms. Support your answer with the help of example.

Ans. Heuristics and algorithms can be distinguished as :

1. Heuristics are like four guides. They point is generally interesting direction.
2. Heuristics may miss the point of interest to particular individual, but solution guaranteed algorithm. Consider all the cases, they do not miss any point of interest to particular individual.
3. Heuristics improves quality of path that are explored, but this is not case with solution guaranteed algorithm.
4. Heuristic algorithm don't guarantee that best solution will be found. Solution guaranted algorithm guarantee that the best solution will be found.
5. Heuristic algorithm usually find a solution close to the best one and they find it fast and easily solution guaranteed algorithm usually find a solution accurate and optimal to best one and find it very accurately.

    Example : Hill climbing, A^* algorithms are the heuristics algorithm, that improves efficiency of search process and BFS and DFS algorithm is the solution guaranteed algorithm.

(c) Solve the water Jug Problem with the capacity of the two Jugs of 3 and 4 liters with no marking in them. Given a water supply with a large storage using these two Jugs, how can you separate 2 litres of water. Also draw the tree, representing the transition of state.

Ans. We are given two jugs, 4 gallon and 3 gallon one. Neither has any measuring markers in it. There is a pump that can be used to fill the jugs with water solution is :

1. (x, y) if $x < 4$	$\rightarrow$	$(4, y)$	fill 4-gallon jug
2. (x, y) if $y < 3$	$\rightarrow$	$(x, 3)$	fill 3-gallon jug
3. (x, y) if $x > 0$	$\rightarrow$	$(x - d, y)$	pour some water out of 4-gallon jug
4. (x, y) if $y > 0$	$\rightarrow$	$(x, y - 0)$	pour some water out of 3-gallon jug
5. (x, y) if $x > 0$	$\rightarrow$	$(0, y)$	empty 4-gallon jug on the ground
6. (x, y)	$\rightarrow$	$(x, 0)$	empty 3-gallon jug
7. (x, y) if $(x + y) \geq 4$ & $y > 0$	$\rightarrow$	$(4, y - (4 - x))$	pour water from 3-gallon jug into 4-gallon jug
8. (x, y) if $(x + y) \geq 3$ & $x > 0$	$\rightarrow$	$(x - (3 - y), 3)$	pour water from 4-gallon jug into 3-gallon jug until full
9. (x, y) if $x + y \leq 4, y > 0$	$\rightarrow$	$(x + y, 0)$	pour all water from 3-gallon jug into 4-gallon jug

10. (x, y) if $(x + y) \le 3$ & $x > 0$	$\rightarrow$	$(0, x + y)$	pour all water from 4-gallon jug into 3-gallon jug
11. $(0, 2)$	$\rightarrow$	$(2, 0)$	pour 2-gallons from 3-gallon jug to 4-gallon jug
12. $(2, y)$	$\rightarrow$	$(0, y)$	empty 2-gallons in 4-gallon jug on ground

State-space diagram

Gallons in 4-gallon jug	Gallons in 3-gallon jug
0	$0 \rightarrow 2$
0	$3 \rightarrow 9$
3	$0 \rightarrow 2$
3	$3 \rightarrow 7$
4	$2 \rightarrow 5$
0	$2 \rightarrow 9$

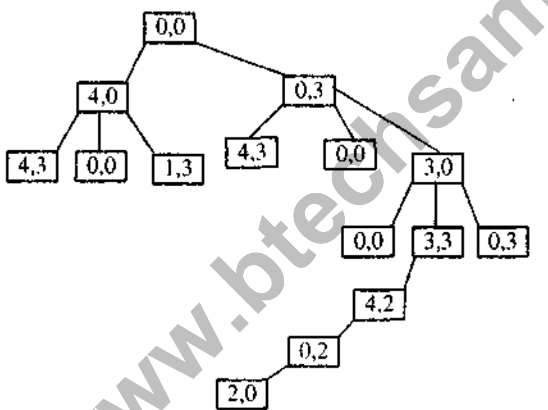


Fig. Tree representing transition of state

(d) Explain the A* algorithm and illustrate the overestimation and underestimation of heuristics.

Ans. The heuristic function used for this algorithm is sum of the cost function and evaluation function.

Heuristic function = Cost function + Evaluation function

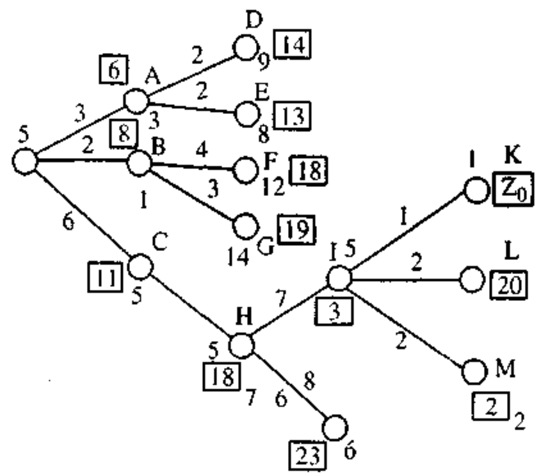
$\Rightarrow$ fitness number

Cost function $\Rightarrow$ how much resources like time energy, money have been spent in reaching particular node from start.

Steps :

1. Put initial node in list START.
2. If (START is empty) or (START = GOAL) terminate search.
3. Remove first node from START. Call this node 9.
4. If (a = GOAL) terminate search with success.
5. Else if 'a' has successors, generate all of them. Estimate fitness number of successors and sort the list by it.
6. Name new list as START 1.
7. Replace START with START 1.
8. Go to step 2.

A sample tree with fitness number used for A* search :

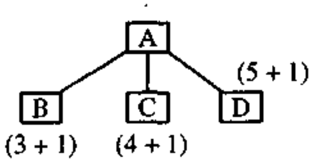


Example : Fitness number of E = (Evaluation function of E) + (Cost function involved from starting node S to node E)

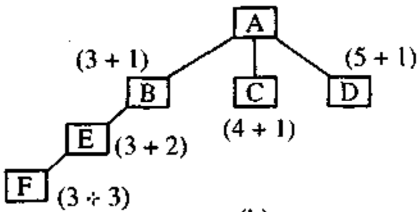
$= 8 + (3 + 2)$

$= 13$

Under estimation of Heuristics : The example to illustrate under estimation of heur'stics, can be given as :



(a)



(b)

In this example, $h'(n)$ underestimates $h(n)$.

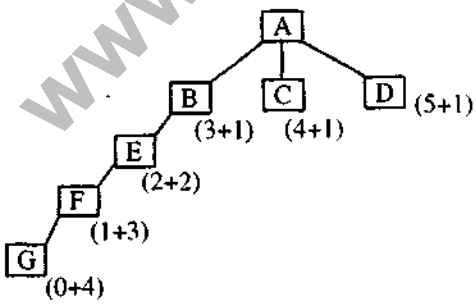
where, $f'(n) = g'(n) + h'(n)$

where, $f'(n) \Rightarrow$ fitness number

$h'(n) \Rightarrow$ estimate of additional cost of getting from current node to goal state.

$g'(n) \Rightarrow$ estimate of getting from initial state to current node.

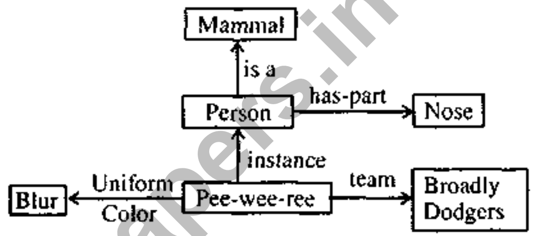
Overestimation of heuristics : Illustration of overestimation of heuristics diagrammatically as an example is given as :



(e) What are semantic Nets? Explain with example how is the knowledge represented.

Ans. In semantic net, information is represented as a set of nodes connected to each other by a set of labelled arcs, which represent relationship among the nodes. This is a natural way to represent relationships, in predicate logic.

Example :



(A semantic network)

For example, some of the areas from the above figure could be represented in logic as :

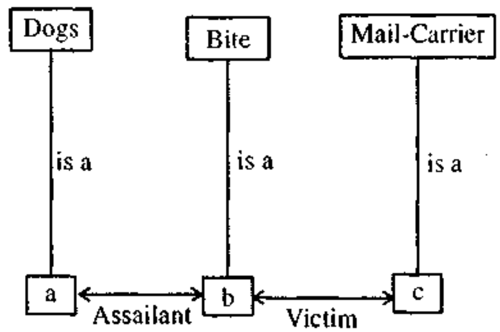
is a (Person, Mammal)

instance (Pee – Wee – Reese, Person)

team (Pee – Wee – Reese, Broklyn – Dodgers)

uniform-colour (Pee – Wee – Reese, Blur)

This technique is particularly useful for representing this contents of a typical declarative sentence that describe several aspects of a particular event. This sentence "The dog bit the mail carrier" can be represented by following semantic net.



(f) Discuss the basic elements of natural language processing.

Ans. Natural language processing can be defined as the automatic processing of human language.

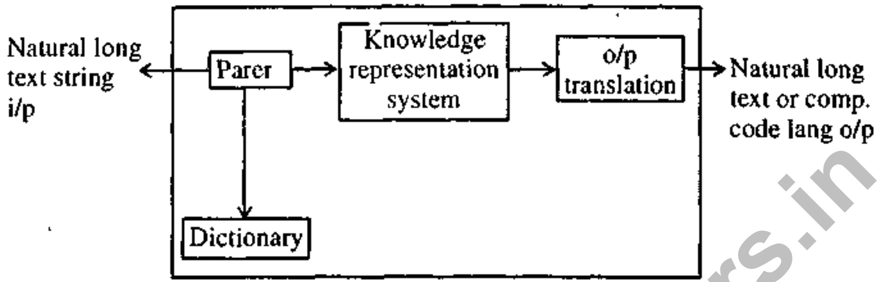


Fig. *The major components of a natural language processing system*

Parser : Parser is the system component that parsers the natural language text string that is given as an input. Then, it forwards it to KRS.

Knowledge representation system : This system is referred as knowledge representation system, which uses various systems and ways for appropriate representation of knowledge.

O/p translation : The final output that come across at the end of natural language processing is under the output translation component.

Dictionary : This is implicit data or stored database in the form of database dictionary, that can be used for parsing in natural language processing.

2. Attempt any TWO of the following :

(a) Describe the difference between language understanding and language generation with suitable examples.

Ans. Natural language generation may be viewed as the opposite of natural language understanding. The difference is :

1. In natural language understanding the system needs to disambiguate the input sentence to produce the machine representation language. In natural language generation, the system needs to make decisions about how to put a concept into word.
2. Natural language generation generates natural language from a machine representation system as a knowledge base or a logical form. Natural language understanding deals with machine reading comprehension, the occurrence of unknown and unexpected features in the input and the need to determine in the input and the need to determine the appropriate syntactic and semantic schemes to apply to it makes it more complex.
3. Application of natural language understands are : News-gathering, data mining, voice-activation, arching and large-scale content analysis.

Application of Natural Language Generation :

1. System that produces textual weather forecasts from weather data.
2. Data-to-text system that generates textual summaries of databases and data set.
3. The natural language understanding system needs a lexicon of the language and a parser and grammar rules to break sentences into an internal representation. The natural language generation needs following stages :
   (i) **Content Determination :** Deciding what information to mention in text.
   (ii) **Text Planning :** Overall organisation of information to convey.

(iii) **Realization :** Creating actual text.

Example of Natural Language Generation : The system that generate from letters. Such systems do not typically involve grammar rules, but may generate a letter to a consumer, i.e. stating a credit card spending limit is about to be reached.

(b) What do you understand by Parsing? Draw Parser tree and define the grammars.

- **A boy ate the Frog.**

Ans. "Parsing : The process of determining the syntactical structure of a sentence is known as parsing" OR "Parsing is the process of analysing a sentence by taking it apart word-by-word and determining its structure from its constituent parts and subparts."

The parsing process is basically the inverse of the sentence generation process.

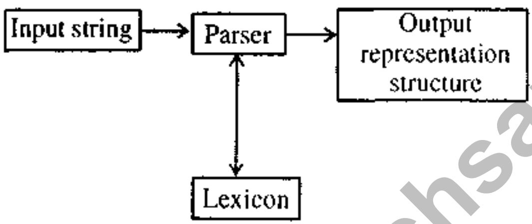


Fig. *Passing an input to create an output sentence*

Some Parsing Techniques : They are following :

1. Transition Network
2. Recursive Transition Network
3. Augmented Transition Network
4. Top Down Parsing
5. Bottom Up Parsing

"A boy ate the frog."

Grammar : G = (V_n, V_t, S, P)

V_n = {S, NP, N, VP, V, ART}

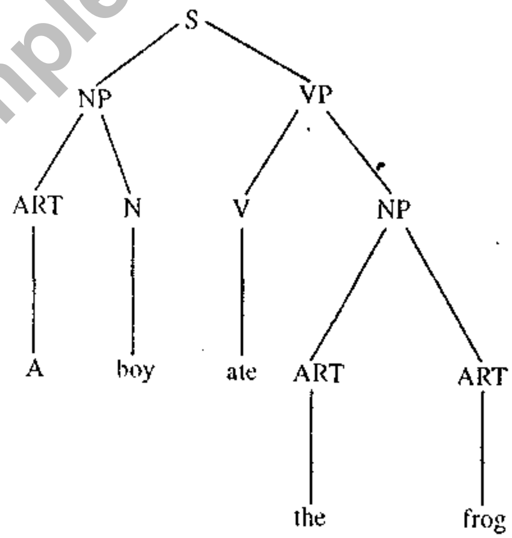
V_t = {boy, frog, ate, a, the}

P :

S → NP VP
NP → ART N
VP → V NP
N → boy | frog
V → ate
ART → the | a

Derivation :

S → NP VP
→ ART N VP
→ A N VP
→ A boy VP
→ A boy V NP
→ A boy ate NP
→ A boy ate ART N
→ A boy ate the N
→ A boy ate the frog



(Passer tree)

(c) What do you understand by term Linguistics? What are the levels of knowledge used in language understanding?

Ans. Linguistics : An understanding of linguistics is not a prerequisite to the study of natural language understanding, but a familiarity with the basics of grammar is certainly important.

We must understand how words and sentences are combined to produce meaningful word strings before we can expect to design successful language understanding systems. In a natural language, the sentence is the basic language element. A sentence is made up of words which express a complete thought. To express a complete thought, a sentence must have a subject and a predicate. The subject is what the sentence is about, and the predicate says something about the subject.

Levels of Knowledge used in Language Understanding : A language understanding program must have considerable knowledge about the structure of the language including what the words are and how they combine into phrases and sentences.

The component forms of knowledge needed for an understanding of natural language are sometimes classified according to the following levels :

1. **Phonological :** This is knowledge which relates sounds to the words we recognise. A phoneme is the smallest unit of sound. Phones are aggregated into word sounds.
2. **Morphological :** This is lexical knowledge which relates to word constructions from basic units called morphemes. A morpheme is the smallest unit of meaning, for example, the construction of friendly from the root friends and suffix ly.
3. **Syntactic :** This knowledge relates to how words are put together or structured to form grammatically correct sentences in the language.
4. **Semantic :** This knowledge is concerned with the meaningful of words and phrases and how they combine to form sentence meaning.
5. **Pragmatic :** This is high-level knowledge which relates to the use of sentences in different contexts and how the context affects the meaning of the sentences.
6. **World :** World knowledge relates to the language a user must have in order to understand and carry on a conversation. It must include an understanding of the other person's beliefs and goals.

The approaches taken in developing language understanding programs generally follow the above levels or stages.

3. Attempt any two of the following :

(a) Explain the Wang's Algorithm and verify the following :

(i) $p \vee q, p \rightarrow s, q \rightarrow s \Rightarrow s$

(ii) $p \rightarrow (q \rightarrow r) \Leftrightarrow (p \wedge q) \rightarrow r.$

Ans. Wang's Algorithm : Wang's algorithm is a mechanistic approach to decide weather an assertion in the proportional calculus is true. Hao Wang published this in 1960. The algorithm involves following steps :

1. Write the premise separated by commas, followed by an arrow (→), and then the theorem to prove.
2. If a formula is negated, remove the "~" and move the formula to the opposite side of the arrow.
3. If any formula on the LEFT has the format (alpha & beta), replace this with "alpha, beta". Similarly any formula on RIGHT with format (alpha [illegible] beta) can be replaced with "alpha, [illegible]
4. [illegible] formula "(alpha ∨ beta)" on the LEFT must be turned into TWO lines, one containing alpha and other beta. Both lines must be proved likewise for AND on the right.
5. If at any point the same formula appears on both sides of an arrow, that line is PROVED.

6. If none of $\vee$ and or ~ is left anywhere in a line, and condition (5) hasn't been met then the line can't be proved.

Note : Wang's algorithm talks about ADD, OR and NOT.

(a) The 'implies' can be converted :

$a \to f \; : \; f \vee \sim a$

(b) XOR can be converted as

$e \oplus d \; : \; (e \vee d) \;\&\; \sim (e \;\&\; d)$

Verification :

(i) $p \vee q, p \to s, q \to s \;\Rightarrow\; s$

Applying substitution,

$p \vee q, s \vee \sim p, s \vee \sim q \;\Rightarrow\; s$

Applying Rule No. 4 So,

$p \vee q, s \vee \sim p, s \Rightarrow s$ (✓) verified

$p \vee q, s \vee \sim p, \sim q \Rightarrow s$ ...(1)

Apply Rule No. 2 on eq. (1)

$p \vee q, s \vee \sim p \Rightarrow s, q$

Applying Rule No. 4

$p \vee q, s \Rightarrow s, q$ (✓) verified

$p \vee q, \sim p \Rightarrow s, q$ ...(2)

Applying Rule No. 2 on eq. (2)

$p \vee q \Rightarrow s, q, p$

Applying Rule No. 4

$p \Rightarrow s, q, p$ (✓) verified

$q \Rightarrow s, q, p$ (✓) verified

(ii) $p \to (q \to r) \;\Leftrightarrow\; (p \wedge q) \to r$

Replace implies with : $a \to b \; : \; b \vee \sim a$

$p \to (r \vee \sim q) \;\Leftrightarrow\; r \vee \sim (p \wedge q)$

$((r \vee \sim q) \vee \sim p) \;\Leftrightarrow\; r \vee \sim (p \wedge q)$

Apply Rule No. 3

$((r \vee \sim q) \vee \sim p) \;\Leftrightarrow\; r, \sim (p \wedge q)$

Apply Rule No. 2

$((r \vee \sim q) \vee \sim p), (p \wedge q) \;\Leftrightarrow\; r$

Apply Rule No. 3

$((r \vee \sim q) \vee \sim p), p, q \;\Leftrightarrow\; r$

Apply Rule No. 4

$(r \vee \sim q), p, q \;\Leftrightarrow\; r$ ...(1)

$\sim p, p, q \Leftrightarrow r$ ...(2)

Evaluating eq. (1)

$(r \vee \sim q), p, q \;\Leftrightarrow\; r$

Apply Rule No. 4

$r, p, q \Leftrightarrow r$ (✓) verified

$\sim q, p, q \Rightarrow r$ ...(3)

Apply Rule 2 on eq. (3)

$p, q \Leftrightarrow r, q$ (✓) verified

Evaluating eq. (2)

$\sim p, p, q \Leftrightarrow r$

Apply Rule 2

$p, q \Leftrightarrow r, p$ (✓) verified

then,

there exists a reputation of $\leftarrow A_1, \ldots, A_m$ with answer θ such that

$\sim (A_1 \wedge \ldots \wedge A_m)\theta$ is an instance of

$(A_1 \wedge \ldots \wedge A_m)\theta$

(b) **Explain the first order Predicate logic (FOL). Explain how well formed formulas (WFF) are defined.**

Rewrite the following sentences in FOL.

(i) Coconut is a biscuit.

(ii) Mary is a child who takes coconut.

(iii) John loves children who take biscuits.

(iv) For a triangle ABC it is given that the sum of the interior angle is 180 degree.

Ans. First Order Predicate Logic : FOPL has one of the most important roles in AI for representation of knowledge.

- It offers the only formal approach to reasoning that has a sound theoretical foundation.
- The structure of FOPL is flexible enough to permit the accurate representation of natural language reasonably well.

In FOPL, statements forms a natural language like English are translated into symbolic structure comprises of :

1. **Constants :** Constants are fixed value terms that belong to a given domain of discourse. They are denoted by numbers, words and capital letters near the beginning of the alphabet such as A, B, C, 5.3, – 21 etc.
2. **Predicates :** Predicate symbols denote relations or functional mappings from the elements of a domain to the values. True of False. For e.g., Brother, king, crown, equal.
3. **Functions :** Function symbols denote relations defined on a domain symbols such as left leg, Father Of, Age Of represent functions.
4. **Variables :** Variables are terms that can assume different values over a given domain such as aircraft type, individuals, x, y, z etc.
5. **Quantifiers :** The two quantifier symbols are $\exists$ (existential quantifier) and $\forall$ (universal quantification).
6. **Connectives :** FOPL uses five connectives : $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\rightarrow$ (implication), $\leftrightarrow$ (equivalence)

Rules for well formed formula : An atomic formula is a wff. If P and Q are wffs, then

(i) $\neg P$

(ii) $P \wedge Q$

(iii) $P \vee Q$

(iv) $P \rightarrow Q$

(v) $P \leftrightarrow Q$, $\forall x\ P(x)$ and $\exists x\ P(x)$

all are wffs :

The above rules states that all wffs are formed from atomic formulas with the proper application of quantifiers and logical connectives.

Some definitions as in the case of propositional logic are :

1. A wff. is said to be valid if it is true under every interpretation. For example $P \vee \neg P$ is valid.
2. A wff which is false under every interpretation is inconsistent or unsatisfiable. For example, $P \wedge \neg P$ is inconsistent.
3. A wff which is not valid (that is false) for some interpretation is invalid.
4. A wff which is not inconsistent (true) for some interpretation is satisfiable.

Sentences in FOL :

(i) Coconut is a biscuit.

BISCUIT (Coconut)

(ii) Mary is a child who takes coconut.

CHILD (Mary) $\wedge$ TAKE (Coconut, Mary)

(iii) John loves children who take biscuits

$\forall x$: CHILD (x) $\wedge$ TAKE (biscuit) $\rightarrow$ LOVES (John, x)

(iv) For a triangle ABC it is given that the sum of the interior angle is 180 degree.

$\forall y\ \forall x$: triangle (y) $\rightarrow$ sum (x, 180) $\wedge$ INTERIOR ANGLE (y)

(c) Explain the Resolution Algorithm and explain the soundness and completeness of resolution algorithms. Prove the following goal with the help of knowledge base.

(i) $p \vee q$

(ii) $\neg q \vee t$

(iii) $\neg p \vee s$

(iv) $\neg s$

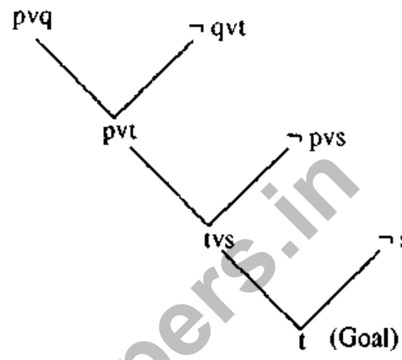
goal : t.

Ans. Resolution Algorithm : In propositional logic, the procedure for producing a proof by resolution of proposition P with respect to a set of axioms F is the following :

Algorithm : Propositional Resolution :

1. Convert all the propositions of F to clause form.
2. Negate P and convert the result to clause form. Add it to the set of clauses obtained in step 1.
3. Repeat until either a contradiction is found or no progress can be made :
   (a) Select two clause. Call these the parent clauses.
   (b) Resolve them together. The resulting clause, called the resolvent, will be the disjunction of all of the literals of both of the parent clauses with the following exception. If these are any pairs of literals L and $\neg$L such that one of the parent clauses contains L and the other contains $\neg$L, then select one such pair one and estimate both L and $\neg$L from the resolvent.
   (c) If the resolvent is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

Numerical : Since given axioms are already in clause form. Therefore,



Soundness : In a formal system $\langle S, L \rangle$ the resolution algorithm L are sound if and only if any sentences which can be derived from $\langle S, L \rangle$ is a logical sentence of $\langle S, L \rangle$.

Completeness : If $\langle S, L \rangle$ is a formal system, then the influence procedure L is complete if and only if any sentence and logically implied by $\langle S, L \rangle$ can be derived using that procedure.

where,

S = set of facts or axioms

L = set of inference rules

For soundness :

Let – P be a definite program

– R or selection function

– Q or R computed answer for goal $\leftarrow A_1, \ldots A_m$

Then,

$$-P_1 = (\forall (A_1 \wedge A_2 \ldots \wedge A_m))\theta$$

For completeness :

Let $P_1 \leftarrow A_1, \ldots A_m$, and R is same before

if $P \,|= \forall (A_1 \wedge \ldots \wedge A_m)\sigma$

4. **Attempt any two of the following :**

(a) **What is an expert system? Explain the following methodologies.**

(i) **Probabilistic techniques**

(ii) **Fuzzy based techniques.**

Ans. Expert System : "An expert system is a set of programs that manipulate encoded knowledge to solve problems in a specialised domain that normally requires human expertise."

An expert system's knowledge is obtained from expert sources and coded in a form suitable for the system to use in its inference or reasoning process. The expert knowledge must be obtained from specialists or other sources of expertise, such as texts, journal articles and the databases. This type of knowledge require much training and experience in some specialised field such as medicine, geology, system configuration or engineering design.

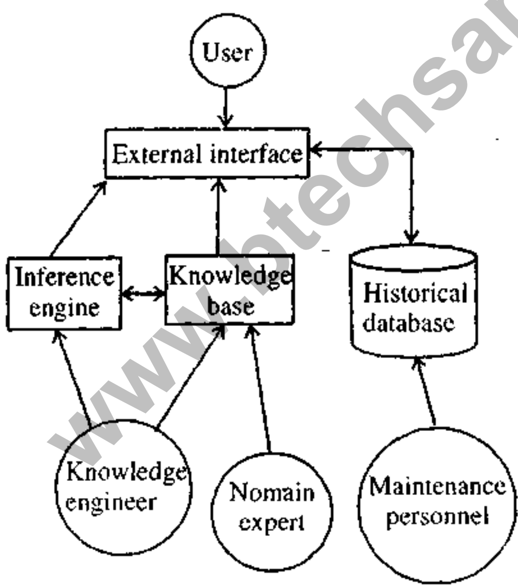


Fig. Personnel involved in expert system development

Characteristics of Expert System :

1. Expert system use knowledge rather than data to control the solution process. Much of the knowledge used is heuristics in nature.
2. The knowledge is encoded of maintenance as an entity separate from the control program.
3. Expert system are capable of explaining how a particular conclusion was reached.
4. Expert systems uses symbolic representations for knowledge like rules, networks and frames.
5. Expert system reason with metaknowledge.

(i) **Probabilistic Techniques :** Probabilistics expert system based on Bayesian networks require initial specification of both qualitative graphical structure and quantitative conditional probability assessments. As data accumulate on real cases, the parameters of the system may adapt, but it is also essential that the initial specifications be monitored with respect to their predictive performance. A range of monitors based on standardized scanning rules that are designed to detect both qualitative and quantitative departures from the specified model is presented. A simulation study demonstrates the efficacy of these monitors at uncovering such departures.

(ii) **Fuzzy Based Techniques : Fuzzy Logic :** It is a super set of conventional (Boolean) logic that has been extended to handle the concept of partial truth-truth values between "completely true" and "completely false". It was introduced by Dr. Lotfi in 1950.

Some of the expert system are based on fuzzy logic instead of Boolean logic. In other words, Fuzzy expert system is a collection of membership functions and rules that are used to reason about data. Unlike conventional expert systems, which are mainly symbolic reasoning engines, fuzzy expert systems are oriented towards numerical processing.

The rules in Fuzzy techniques based expert system are similar to the following :

if x is low any y is high then z = medium

where x and y are input variables and z is output variable, low is G membership function defined on x, high is a membership function defined on y and medium is a membership on z. This part of rule between 'if' and 'then' is the rules premise or antecedent. This is a fuzzy logic expression that describes to what degree the rule is applicable. The part of the rule following the 'then' is the rules conclusion or consequent. This part of the rule assigns a membership function to each of one or more output variables.

A typical fuzzy expert system has more than one rule. The entire group of rules is collectively known as a Rule base or Knowledge base.

Application that are using Fuzzy based expert system are :

1. Linear and Non-linear control
2. Pattern Recognition
3. Financial Systems

(b) Explain the Dempster Shafer theory for uncertainty management in exert System with suitable example.

Ans. Dempster Shafer Theory : This approach considers sets of propositions and assigns to each of them an interval.

[Belief, Plausibility]

Belief means the strength of the evidence in favour of a set of prepositions. It ranges from 0 (indicating no evidence) to 1 (denoting certainty).

Plausibility (PI) is defined as :

$$P(s) = P - Bel(\neg s)$$

It also ranges from 0 to 1 and measures the extent to which evidence in favour of $\neg S$ leaves room for belief in s. In particular, if we leave certain evidence in favour of $\neg S$, $Bel(\neg s)$ will be 1 and P(s) will be 0. This will tell us that the only possible value for Bel(s) is also 0.

The frame of discernment can be written as θ, which consists of the set {All, Flu, Cold, Pneu}.

Our goal is to attach some measure of belief to elements of θ. However, not all evidence is directly supportive of individual elements. Often it supports sets of elements (to subsets of θ). Example, in our diagnostic problem, fever must support {Flu, Cold, Pneu}. In addition, since the elements of θ are mutually exclusive, evidence in favour of some may have an affect on our belief in the others.

Dempstic Shafer theory lets us handle interactions by manipulating sets of hypothesis directly.

The key function we use is a probability density function, which we denote as m. The function m is defined not just for elements of θ but for all subsets of it (including singleton subsets, which correspond to individual 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^n subsets of θ. We must assign m so that the sum of all the m values assigned to the subsets of θ is 1.

We define m as :

{θ} (1.0)

All the other values of m are thus 0.

Now, suppose we acquire a piece of evidence 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.

We update m as follows :

{Flue, Cold, Pneu}	(0.6)
{θ}	(0.4)

At this point, we have assigned to the set {Flu, Cold, Pneu} the appropriate belief.

Having defined m, we can now define Bel(P) for a set P as the sum of the values of m for P and for all of its subsets.

Thus Bel (P) is our overall belief that the correct lies somewhere in the set P.

- Basic probability assignment :

  $m : 2D \rightarrow [0, 1]$

  m_1 : {Flu, Cold, Pneu} 0.6
- Belief :

  $Bel : 2D \rightarrow [0, 1]$

  $Bel\ (s) = \sum X \subseteq S \text{ in } (X)$
- Plausibility :

  $Pl\ (p) = 1 - Bel\ (\neg P)$

  $P : [Bel(P), Pl(P)]$
- Combination rules : m_1 and m_2

  $\sum X \cap Y = Zm_1(X).m_2(Y)$

  $m(Z) = 1 - \sum X \cap Y = \phi m_1(X).m_2(Y)$

  D 0.4

Example :

m_1 : {Flue, Cold, Pneu} 0.6 D 0.4

m_2 : {All, Flue, Cold} 0.8 D 0.2

	{A, F, C} 0.8	D 0.2
{F, C, P} 0.6	{F, C} 0.48	{F, C, P} 0.12
D 0.4	{A, F, C} 0.32	D 0.08

(c) Discuss the advantage of expert system architecture based on decision tree and production rules.

Ans. Advantages of Expert Systems : Expert systems are being developed for a wide variety of domains. What makes one to go in for ES is the advantages he gets out of it. Some of its advantages are :

1. Expert system have in their knowledge base vast quantities of domain specific knowledge. The knowledge encompasses conventional knowledge obtained from printed material and expert's private knowledge, the heuristics.
2. In an expert system, one can view the entire reasoning process. The reasoning process is not only transparent but also provides answers for questions like "why and how".
3. In the expert system, the knowledge engineer after eliciting the knowledge from the expert codes it into a machine understandable form. Today's expert system accepts a statement from the user directly and converts it into an understandable form by itself. This facility has minimised human intervention and has become intimate the domain experts who might not be a very good computer literate.
4. Expert systems of today have mechanisms for handling uncertain and contradictory evidence. Domain specific experts are satisfied that these mechanisms exactly picturise their line of reasoning.
5. Human experts, under stress or in bad mood or when time is critical, either wake default assumptions or forget relevant factors. Since, ES does not have these characteristics of stress or moods, they do not make default assumptions or forget relevant factors. Hences can have greater reliability.
6. Expert systems have the major advantage of increased accessibility which cannot be imagined in case of human experts.
7. Expert systems takes considerable amount of time to transfer an amateur into a full fledged experts. The time for duplication of an expert system is very short.
8. An ES plays three major roles :

   Role of an archive, a problem solver, and a tutor.

An human expert who is a good problem solver need not to be a good tutor.

9. Expert systems are highly advantageous in interdisciplinary domains where multiple experts are needed.

10. Another advantage of high reliability is that factors which used more importance are always given higher priority.

5. Attempt any two of the following :

(a) Describe the various designing steps of Pattern Recognition System in detail with suitable example.

Ans.

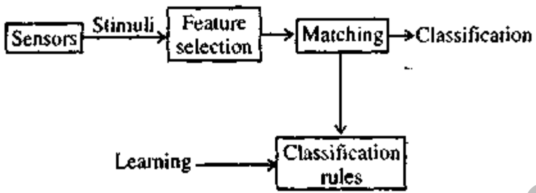


Fig. *Pattern Recognition Process*

Various designing steps in pattern recognition system :

Step 1 : Stimuli produced by objects are perceived by sensory devices. The more prominent attributes (such as size, shapes, colour and texture) produce the strongest stimuli. The values of these attributes and their relations are used to characterise an object in the form of pattern vector X, as a string generated by some grammar, as a classification tree, a description graph, or some other means of representation. The range of characteristics attribute values is known as the measurement space M.

Step 2 : A subset of attributes whose values provide cohesive object grouping or clustering, consistent with some goals associated with the object classifications, are selected. Attributes selected are those which produce high interclass and low interclass grouping. This subset represents a reduction in the attribute space dimensionality and hence simplifies the classification process. The range of the subset of attribute values is known as the feature space F.

Step 3 : Using the selected attribute values, object or class characterisation models are learned by forming generalized phototype descriptions, classification rules, or decision functions. Those models are stored for subsequent recognition. The range of the decision function values or classification rules is known as the decision space D.

Step 4 : Recognition of familiar objects is achieved through application of the rules learned in step 3 by comparison and matching of object features with the stored model. Refinements and adjustment can be performed continually thereafter to improve the quality and speed of recognition.

(b) Describe the symbolic description and structured description and their advantages and disadvantages over system design.

Ans. Symbolic Description : It is important to distinguish between the mechanism for building the meaning of the fake word/phrase and the mechanism that uses the meaning to build a question with the word/phrase as an answer.

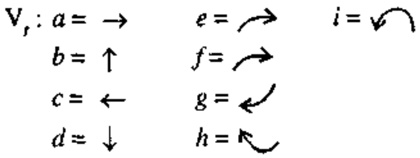
For example, the following questions use the same meaning for the fake word 'trunkquillizer', but refer to that meaning in different types or ways :

- What do you use to sedate an elephant?
- What do you call elephant sedative?
- What kind of medicine do you give to a stressed-out elephant?

On the other hand, these questions are all put together in the same way, but from different fake meanings of the fake word.

- What do you use to sedate an elephant?
- What do you use to sedate a piece of luggage?
- What do you use to medicate a nose?

Syntactic Classification : The syntactic recognition approach is based on the uniqueness of syntactic "structure" among the object classes. With this approach a grammar is defined object description. Instead of defining the grammar is terms of an alphabet of characters or terminal words, the vocabulary is based on shape primitives. For example, the object shown in figure could be defined using the grammar G (V_n, V_t, P, S), where the terminal V_t consist of the following shape primitives :



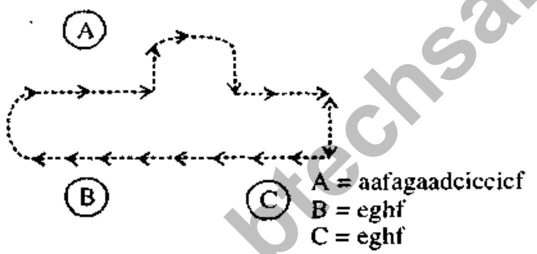


Fig. *Syntactic characteristics of objects*

Using syntactic analysis, that is parsing and analysing the string structures, classification is accomplished by assigning an object to class C_i when the string describing it has been generated by the grammar G_1.

Advantages : Symbolic Description

- It supports abstraction of information
- It is helpful in organisations like in defence, CBI, etc.

Advantages : Syntactic Description

- It is a simplest form of information recognition.
- It is more flexible than other recognition methods.

Disadvantages : Symbolic Description

- Sometimes it is difficult to understand the fake message.
- It is more time taken.

Disadvantages : Syntactic Description

- When pattern are noisy or subject to random fluctuations, ambiguities may occur since pattern belong to different classes may appear to be the same.

(c) Explain the various list manipulation functions of LISP with example.

Ans. List Manipulation Primitives : The purpose of list manipulation primitives is to

(i) Create a new list

(ii) Modify our existing list with addition, deletion or replacement of an atom.

(iii) Extraction of portions of a list.

- SETF Primitive (or SETQ Primitive)
  - Assign a value to a symbol (SET field).
  - Returns the 2nd argument as its value.
  - Side Effect : Assigning the value of its 2nd argument to its 1st-argument.

  For example, suppose we want the value of E to be 272. We can do this as follows :

  1. * (SETF E 2.72)

     2.72

     * E

     2.72

     *

- SETF accepts multiple symbol value pairs but only the value of the final argument is returned.
- **List Selectors :** These are the basic list selectors :

Selector	Purpose	Example
FIRST	Return 1st element of the list	(2)
REST	Return list after removing the first element.	(3)
LAST	Return list after removing all but one element.	(4)
BUTLAST	Return list after removing the last *n* elements of a list, with the 2nd argument determining the exact number.	(5)
NTHCDR	Return list after removing the first *n* elements of a list, with the 1st argument determining the exact number.	(6)
LENGTH	Return the number of top-level elements in a list.	(7)

For example :

1. * (self mylist' (Fast computers are nice))
   (FAST COMPUTERS ARE NICE)
2. * (first mylist)
   FAST
3. * (rest mylist)
   (COMPUTER ARE NICE)
4. * (last mylist)
   (NICE)
5. * (butlast mylist 2)
   (FAST COMPUTERS)
6. * (nthcdr 2 mylist)
   (ARE NICE)
7. * (length mylist)
   4
   *

- **List Constructors—CONSTRUCT :** Takes an expression and a list and returns a new list whose 1st element is the expression and whose remaining elements are those of the old list.

For example :

* (CONS' A' (BC))

(ABC)

Assuming L's value is (AB) and M's value is ((LM) (XY)), then :

* (CONS LM)

((AB) (LM) (XY))

* (cons LL)

((AB) AB)

* (cons '(a)' (bc))

((AB) BC)

you will often combine SETF and CONS this way :

(SETF < name of a list >

(cons < new element > < name of the list >))

*** List Constructors—APPEND :** Combines the elements of all lists supplied as arguments :

For example :

* (APEND '(ABC)'(XYZ))

```
 |   |   |   |   |   |
 |   |   |   |   |   |
 v   v   v   v   v   v
(A   B   C   X   Y   Z)
```

Compare the result with what you would get with CONS :

* (CONS' (ABC) (XYZ))

((ABC) XYZ)

- APPEND makes a list out of all the elements in its arguments.
- CONS adds its first argument to its second argument, a list.

- **List Constructors—REVERSE :** Reverse the order of the top-level elements of a list.

For example :

- (REVERSE (ABC))
  (CBA)

Note : However, that REVERSE does not turn the individual elements of a list around when those elements are lists :

For example :

- (REVERSE '((AB)(LM) (XY)))
  ((XY) (LM) (AB))