AI Sessional-2 Important Questions

1.Consider the following sentences, translate into formulas in predicate logic:

(a) John likes all kinds of food. (b) Apples are food. (c) Chicken is food. (d) Anything anyone eats and isn't killed by is food. (e) Bill eats peanuts and is still alive. (f) Sue eats everything Bill eats

Logical form

- John likes all kinds of food == $\forall x$ *food (x)* $\Rightarrow$ *eats(John, x)*
- Apples are food == *food(apples)*
- Chicken is food == *food(chicken)*
- Anything anyone eats and isn't killed by is food == $\forall x,y$ *eats(x,y)* $\wedge \neg$*killed(x)* $\Rightarrow$ *food(y)*
- Bill eats peanuts, and is still alive == *eats(Bill,Peanuts)* $\wedge$ $\neg$*killed(Bill)(here we assume alive means not killed)*
- Sue eats everything that Bill eats == $\forall x$ *eats (Bill,x)* $\Rightarrow$ *eats(Sue,x)*

2.Discuss about semantic tableaux system.

semantic tableax system

the set of rules are applied systematically on a formula or set of formulae to establish its consistency or inconsistency.

- **RULES**
- Let α and β be any two formulae.

Rule 1: A tableau for a formula $(\alpha \wedge \beta)$ is constructed by adding both α and β to the same path (branch). This can be represented as follows:

$$\begin{array}{c} \alpha \wedge \beta \\ | \\ \alpha \\ | \\ \beta \end{array}$$

- *Semantic tableau*
  - binary tree constructed by using semantic rules formula as a root
- Assume α and β be any two formulae.

Interpretation: $\alpha \wedge \beta$ is true if both α and β are true

Rule 2: A tableau for a formula $\sim(\alpha \wedge \beta)$ is constructed by adding two alternative paths one containing $\sim\alpha$ and other containing $\sim\beta$

$$\begin{array}{ccc} & \sim(\alpha \wedge \beta) & \\ \swarrow & & \searrow \\ \sim\alpha & & \sim\beta \end{array}$$

Interpretation: $\sim(\alpha \wedge \beta)$ is true if either $\sim\alpha$ or $\sim\beta$ is true

Cont...

Rule 3: A tableau for a formula $(\alpha \vee \beta)$ is constructed by adding two new paths one containing α and other containing β.

$$\begin{array}{ccc} & \alpha \vee \beta & \\ \swarrow & & \searrow \\ \alpha & & \beta \end{array}$$

Interpretation: $\alpha \vee \beta$ is true if either α or β is true

Rule 4: A tableau for a formula $\sim(\alpha \vee \beta)$ is constructed by adding both $\sim\alpha$ and $\sim\beta$ to the same path. This can be expressed as follows:

$$\begin{array}{c} \sim(\alpha \vee \beta) \\ | \\ \sim\alpha \\ | \\ \sim\beta \end{array}$$

Rule 5: Semantic tableau for $\sim\sim\alpha$

$$\begin{array}{c} \sim\sim\alpha \\ | \\ \alpha \end{array}$$

Rule 6: Semantic tableau for $\alpha \rightarrow \beta$

$$
\begin{array}{ccc}
 & \alpha \rightarrow \beta & \\
\sim\alpha & & \beta
\end{array}
$$

Rule 7: Semantic tableau for $\sim(\alpha \rightarrow \beta)$

$$
\begin{array}{c}
\sim(\alpha \rightarrow \beta) \\
| \\
\alpha \\
| \\
\sim\beta
\end{array}
$$

Semantic Tableaux System...

Aditya Engineering College (A)

Rule 8: Semantic tableau for $\alpha \leftrightarrow \beta$ $\alpha \leftrightarrow \beta \cong (\alpha \wedge \beta) \vee (\sim\alpha \wedge \sim\beta)$

$$
\begin{array}{ccc}
 & \alpha \leftrightarrow \beta & \\
\alpha \wedge \beta & & \sim\alpha \wedge \sim\beta
\end{array}
$$

Rule 9: Semantic tableau for $\sim(\alpha \leftrightarrow \beta)$

$\sim(\alpha \leftrightarrow \beta) \cong (\alpha \wedge \sim\beta) \vee (\sim\alpha \wedge \beta)$

$$
\begin{array}{ccc}
 & \sim(\alpha \leftrightarrow \beta) & \\
\alpha \wedge \sim\beta & & \sim\alpha \wedge \beta
\end{array}
$$

3.Extended semantic network.

Extended Semantic Network (ESNet) combines the advantages of both logic and semantic network.

In the ESNet, terms are represented by nodes similar to Sem Net.

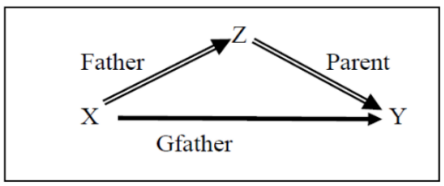
Conclusions and *conditions* in clausal form are represented by different kinds of arcs.

- Conditions are drawn with two lines $\Longleftarrow$ and conclusions are drawn with one heavy line $\longleftarrow$.

  Example:

  - Represent 'grandfather' definition

    Gfather(X, Y) ← Father(X, Z), Parent(Z, Y) in ESNet.

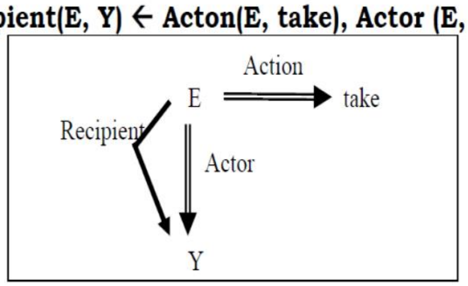


- The inference rule such as "an actor of taking action is also the recipient of the action" can be easily represented in clausal logic as:

  - Here E is a variable representing an event where an action of taking is happening).

Recipient(E, Y) ← Acton(E, take), Actor (E, Y)

ESNet



Example **Extended Semantic Network**

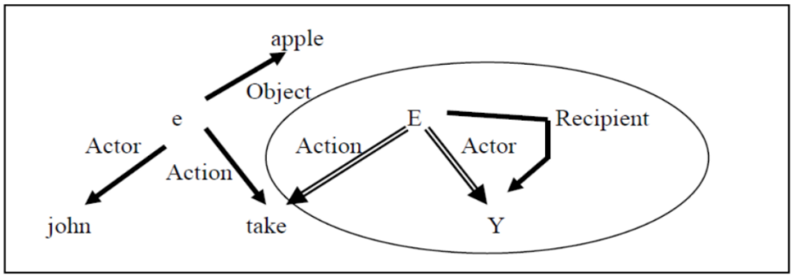
- Represent the following clauses of Logic in ESNet.

Recipient(E, Y) ← Acton(E, take), Actor (E, Y)

Object (e, apple).

Action(e, take).

Actor (e, john) .



4.What is resolution refutation. Outline the conversion formula in PL to transform into equivalent CNF representation.

Resolution refutation is another simple method to prove a formula by contradiction. Resolution is used, if there are various statements are given, and we need toprove a conclusion of those statements

Here negation of goal to be proved is added to given set of clauses.

It is shown then that there is a refutation in new set using resolution principle.

Conversion of a Formula to its CNF

- Eliminate $\rightarrow$ and $\leftrightarrow$ by using the following equivalence laws.

$$P \rightarrow Q \cong \sim P \vee Q$$

$$P \leftrightarrow Q \cong (P \rightarrow Q) \wedge (Q \rightarrow P)$$

- Eliminate double negation signs by using

$$\sim\sim P \cong P$$

- Use De Morgan's laws to push ~ (negation) immediately before atomic formula.

  $\sim(P \wedge Q) \cong \sim P \vee \sim Q$

  $\sim(P \vee Q) \cong \sim P \wedge \sim Q$

- Use distributive law to get CNF.

  $P \vee (Q \wedge R) \cong (P \vee Q) \wedge (P \vee R)$

- We notice that CNF representation of a formula is of the form
  - $(C_1 \wedge \ldots \wedge C_n)$, where each C_k, $(1 \leq k \leq n)$ is a clause that is disjunction of literals.

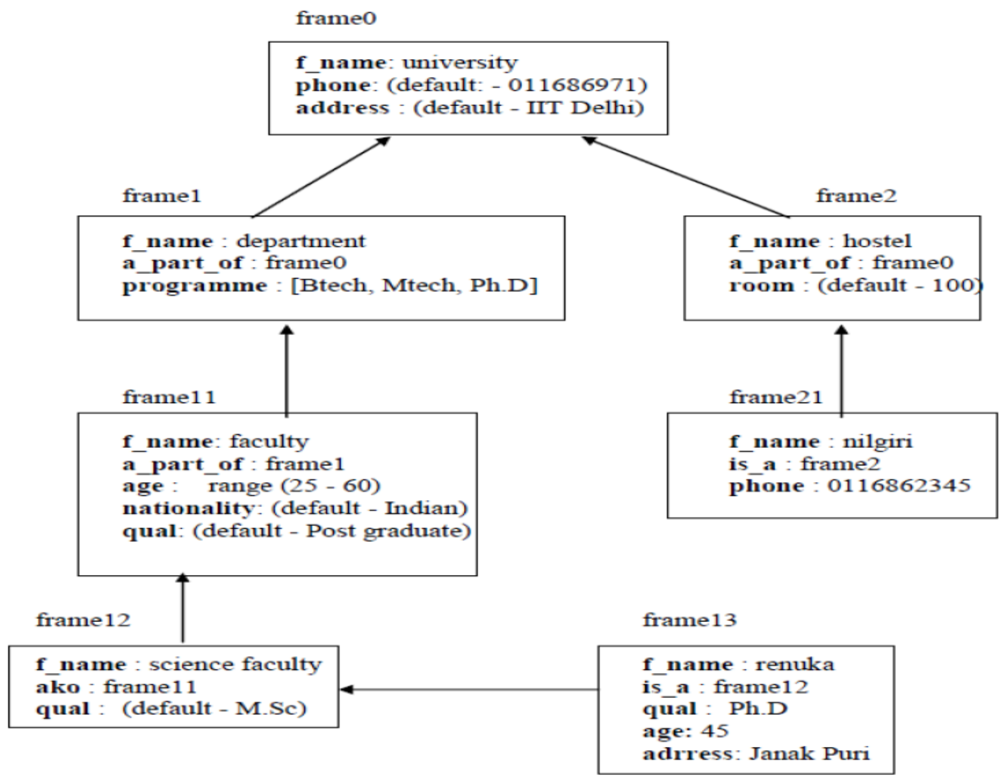
5.Describe about extended semantic network.

Refer question no:3

6.Explain knowledge representation using frames.

-Frames are more structured form of packaging knowledge,
-Used for representing objects, concepts etc.
-Frames are organized into hierarchies or network of frames.
A frame may have any number of slots needed for describing object. e.g.,

faculty frame may have name, age, address, qualification etcas slot names.
Each frame includes two basic elements : slots and facets.
Each slot may contain one or more facets (called fillers) which may take many forms such as:
value ,default ,range,demons and other .



7.Explain knowledge representation in semantic network with a suitable example.

Knowledge representation using Semantic Network

We can represent knowledge using a structure called Semantic Network

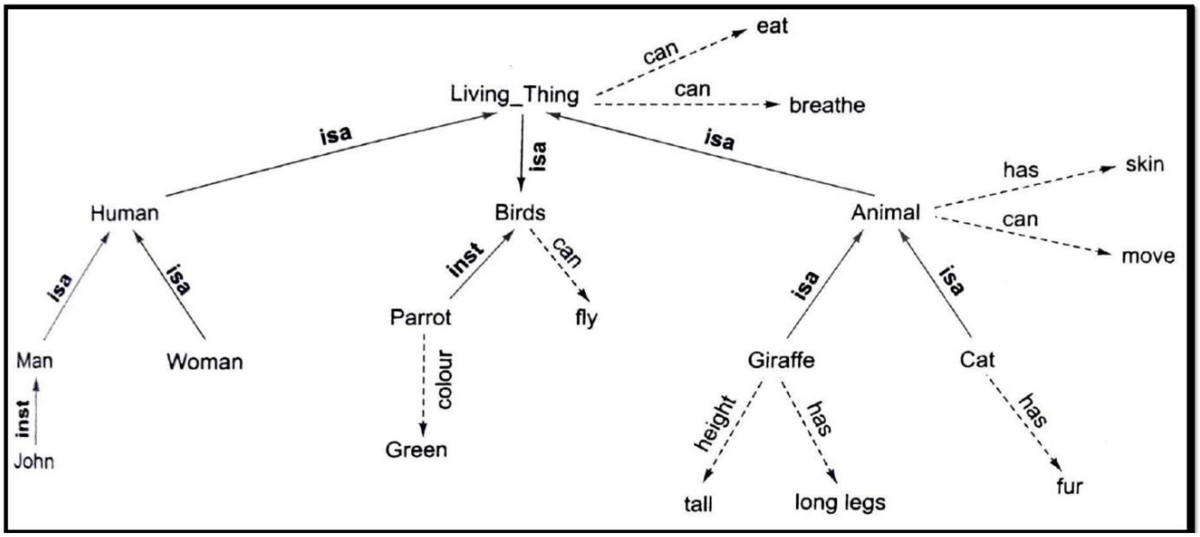
•Nodes represent concepts or objects and arcs represent relation between two concepts

•It can be represented as a directed graph

The meaning of a concept is derived from its relationship with other concepts

•Foreg.

Every human, animal and birds are living things who can breathe and eat. All birds can fly. Every man and woman are human who have two legs. A cat has fur and is an animal. All animals have skin and can move. A giraffe is an animal and has long legs and is tall. A parrot is a bird and is green in colour. John is a man.



- Two types of relations in the figure are:
  - **isa :** connects two classes, where one concept is a kind or subclass of the other concept
  - **inst:** relates specific members of a class
- Other relations such as {can, has, colour, height} are known as property relations. These have been represented by dotted lines pointing from the concept to its property.
- In this structure, property inheritance is easily achieved.
- For example, the query "Does a parrot breathe?" can be easily answered "yes" even though this property is not directly associated with a parrot.
- The information encoded can be represented as a directed graph as shown below.

8.Explain knowledge representation using frames with an example.

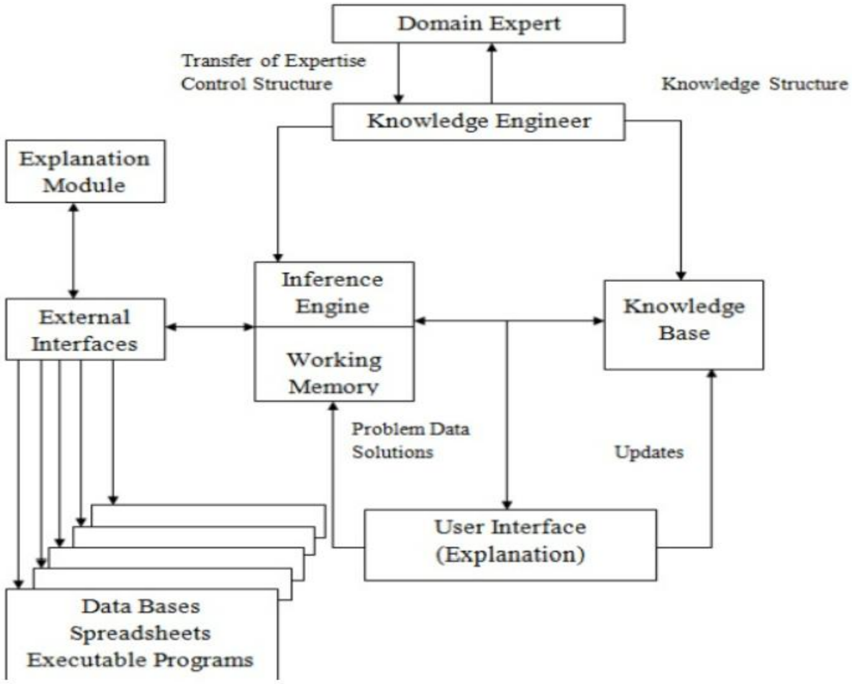
Refer question no:6

9.Illustrate rule based expert system architecture

A rule based expert system is one in which knowledge base is in the form of rules and facts.

- ● It is also called *production system*.

- Knowledge in the form of rules and facts is most popular way in designing expert systems.



User Interface: It is the mechanism by which the user and the expert system communicate with each other

Explanation Module: The explanation module explains the reasoning of the system to a user. It provides the user with an explanation of the reasoning process when requested.

Knowledge Base: In rule based architecture of an expert system, the knowledge base is the set of production rules.

Inference Engine: The inference engine accepts user input queries and responses to questions through the I/O interface

10.Explain Dempster-Shafer theory.

- It is a mathematical theory of evidence.
- It allows one to combine evidence from different sources and arrive at adegree of belief.
- Belief function is basically a generalization of the Bayesian theory ofprobability.
- It also uses numbers in the range [0, 1] to indicate amount of belief in ahypothesis for a given piece of evidence.

Dempster Theory Formalism

- Let *U* be the *universal set* of all hypotheses, propositions, or statements under consideration.
  - The power set P(U), is the set of all possible subsets of *U*, including the empty set represented by ϕ.
  - The theory of evidence assigns a belief mass to each subset of the power set.
- A function m: P(U) $\rightarrow$ [0,1] is called a *basic belief assignment* (BBA) function. It satisfies the following axioms:
  - $m(\phi) = 0$; $\sum m(A) = 1, \forall A \in P(U)$
- The value of m(A) is called *mass assigned to A* on the unit interval.

- It makes no additional claims about any subsets of A, each of which has, by definition, its own mass.

Dempster's Rule of Combination

$$m3(\phi) = 0$$

$$m3(C) = \frac{\sum_{A \cap B = C} (m1(A) * m2(B))}{1 - \sum_{A \cap B = \phi} (m1(A) * m2(B))}$$

- This belief function gives new value when applied on the set $C = A \cap B$.
- The combination of two belief functions is called the *joint mass*.
  - Here m3 can also be written as (m1 o m2).
- The expression $[\sum_{A \cap B = \phi} (m1(A) * m2(B))]$ is called normalization factor.
  - It is a measure of the amount of conflict between the two mass sets.
- The normalization factor has the effect of completely ignoring conflict and attributing any mass associated with conflict to the null set.

11.Explain about Bayesian Belief network.

It is a acyclic (with no cycles) directed graph where the nodes of the graph represent evidence or hypotheses and arc connecting two nodes represents dependence between them.

- If there is an arc from node X to another node Y (i.e., X $\rightarrow$Y), then X is called a *parent* of Y, and Y is a *child* of X.
- The set of parent nodes of a node X_i is represented by parent_nodes(X_i).
- Joint probability for 'n' variables (dependent or independent) is computed as follows.

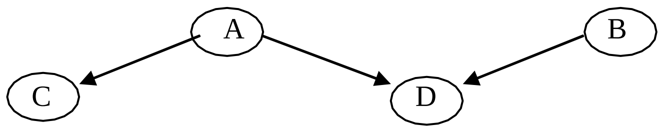
$$P(X_1, \ldots, X_n) = P(X_n \mid X_1, \ldots, X_{n-1}) * P(X_1, \ldots, X_{n-1})$$

- In Bayesian Network, the joint probability distribution can be written as the product of the local distributions of each node and its parents such as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{parent\_nodes}(X_i))$$

- If node Xi has no parents, its probability distribution is said to be unconditional and it is written as P(Xi) instead of P(Xi | parent_nodes(Xi)).
- Nodes having parents are called conditional.
- If the value of a node is observed, then the node is said to be an evidence node.
- Nodes with no children are termed as hypotheses node and nodes with no parents are called independent nodes.
- The following graph is a Bayesian belief network.
    - Here there are four nodes with {A, B} representing evidences and {C, D} representing hypotheses.

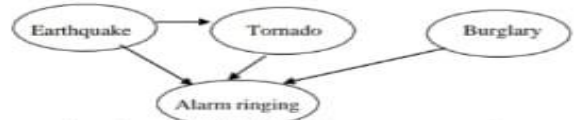
- A and B are unconditional nodes and C and D are conditionalnodes



Bayesian Belief Network

Example of Simple B-Network:

- Suppose that there are three events namely earthquake, burglary or tornado which could cause ringing of alarm in a house.
- This situation can be modeled with Bayesian network as follows.
- All four variables have two possible values T (for true) and F (for false).
  - Here the names of the variables have been abbreviated to A = *Alarm*, E = *Earthquake*, and B = *Burglary* and T = *Tornado*.



- Table contains the probability values representing complete Bayesian belief network. Prior probability of 'earthquake' is 0.4 and if it is earthquake then probability of 'tornado' is 0.8. and if not then the probability of 'earthquake' is 0.5.

Conditional Probability Tables

P(E)	P(B)
0.4	0.7

E	P(Tor)
T	0.8
F	0.5

E	B	Tor	P(A)
T	T	T	1.0
T	T	F	0.9
T	F	T	0.95
T	F	F	0.85
F	T	T	0.89
F	T	F	0.7
F	F	T	0.87
F	F	F	0.3

- The joint probability is computed as follows:

P(E, B, T, A) = P(A| E, B, T) * P(T|E) * P(E) * P(B)

= 1.0 * 0.8 * 0.4 * 0.7 = 0.214

- Using this model one can answer questions using the conditional probability formula as follows:
  - "What is the probability that it is earthquake, given the alarm is ringing?" P(E|A)
  - "What is the probability of burglary, given the alarm is ringing?" P(B|A)
  - "What is the probability of ringing alarm if both earthquake and burglary happens?" P(A|E, B)

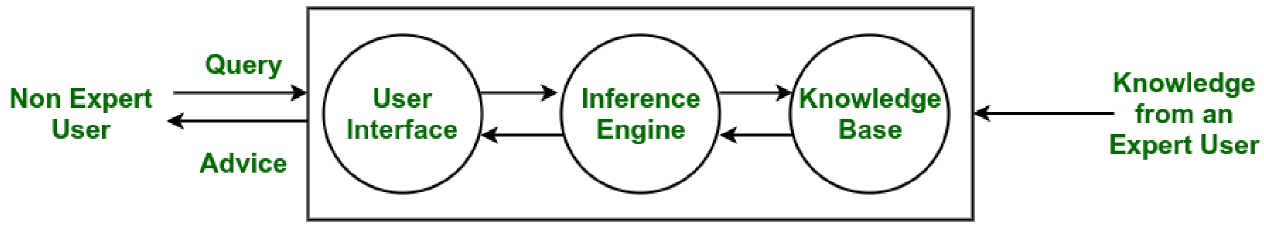
Advantages of Bayesian Belief Network:

- It can easily handle situations where some data entries are missing as this model encodes dependencies among all variables.
- It is intuitively easier for a human to understand direct dependencies than complete joint distribution.
- It can be used to learn causal relationships.
- It is an ideal representation for combining prior knowledge (which often comes in causal form) and data because the model has both causal and probabilistic semantics.

Disadvantages of Bayesian Belief Network:

- The probabilities are described as a single numeric point value. This can be a distortion of the precision that is actually available for supporting evidence.
- There is no way to differentiate between ignorance and uncertainty. These are distinct two different concepts and be treated as such.
- The quality and extent of the prior beliefs used in Bayesian inference processing are major shortcomings.

12.Explain the architecture of an expert system.



- **Knowledge Base –**
  The knowledge base represents facts and rules. It consists of knowledge in a particular domain as well as rules to solve a problem, procedures and intrinsic data relevant to the domain.
- **Inference Engine –**
  The function of the inference engine is to fetch the relevant knowledge from the knowledge base, interpret it and to find a solution relevant to the user's problem. The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts. Inference engines can also include an explanation and debugging abilities.
- **Knowledge Acquisition and Learning Module –**
  The function of this component is to allow the expert system to acquire more and more knowledge from various sources and store it in the knowledge base.
- **User Interface –**
  This module makes it possible for a non-expert user to interact with the expert system and find a solution to the problem.
- **Explanation Module –**
  This module helps the expert system to give the user an explanation about how the expert system reached a particular conclusion.

CSE QUESTIONS:

1.Discuss about the rules of natural deduction system.

Natural Deduction System

- ND is based on the set of few deductive inference rules.
- The name natural deductive system is given because it mimics the pattern of natural reasoning.
- It has about 10 deductive inference rules.

Conventions:

- E for Elimination, I for Introducing.
- P, P_k , $(1 \leq k \leq n)$ are atoms.
- α_k, $(1 \leq k \leq n)$ and β are formulae.

Natural Deduction System

ND RULES:

Rule 1: I-$\wedge$ (Introducing $\wedge$)

I-$\wedge$: If $P_1, P_2, \ldots, P_n$ then $P_1 \wedge P_2 \wedge \ldots \wedge P_n$

Interpretation: If we have hypothesized or proved P_1, P_2, … and P_n , then their conjunction $P_1 \wedge P_2 \wedge \ldots \wedge P_n$ is also proved or derived.

Rule 2: E-$\wedge$ (Eliminating $\wedge$)

E-$\wedge$: If $P1 \wedge P2 \wedge \ldots \wedge Pn$ then Pi $(1 \leq i \leq n)$

Interpretation: If we have proved $P1 \wedge P2 \wedge \ldots \wedge Pn$, then any Pi is also proved or derived. This rule shows that $\wedge$ can be eliminated to yield one of its conjuncts.

Rule 3: I-V (Introducing V)

I-V : If P_i $(1 \leq i \leq n)$ then $P_1 V P_2 V \ldots V P_n$

Interpretation: If any Pi $(1 \leq i \leq n)$ is proved, then $P_1 V \ldots V P_n$ is also proved.

Natural Deduction System

ND RULES…

Rule 4: E-V (Eliminating V)

E-V : If $P_1 V \ldots V P_n$, $P_1 \rightarrow P$, … , $P_n \rightarrow P$ then P

Interpretation: If $P_1 V \ldots V P_n$, $P_1 \rightarrow P$, … , and $P_n \rightarrow P$ are proved, then P is proved.

Rule 5: I-$\rightarrow$ (Introducing $\rightarrow$)

I-$\rightarrow$: If from $\alpha_1, \ldots, \alpha_n$ infer β is proved then $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ is proved

Interpretation: If given α_1, α_2, … and α_n to be proved and from these we deduce β then $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ is also proved.

Rule 6: E-$\rightarrow$ (Eliminating $\rightarrow$) - *Modus Ponen*

E-$\rightarrow$: If $P_1 \rightarrow P$, P_1 then P

Natural Deduction System

ND RULES…

Rule 7: I-$\leftrightarrow$ (Introducing $\leftrightarrow$)

I-$\leftrightarrow$: If $P_1 \rightarrow P_2$, $P_2 \rightarrow P_1$ then $P_1 \leftrightarrow P_2$

Rule 8: E-$\leftrightarrow$ (Elimination $\leftrightarrow$)

E-$\leftrightarrow$: If $P_1 \leftrightarrow P_2$ then $P_1 \rightarrow P_2$, $P_2 \rightarrow P_1$

Rule 9: I-$\sim$ (Introducing $\sim$)

I-$\sim$: If from P infer $P_1 \wedge \sim P_1$ is proved then $\sim P$ is proved

Rule 10: E-$\sim$ (Eliminating $\sim$)

E-$\sim$: If from $\sim P$ infer $P_1 \wedge \sim P_1$ is proved then P is proved

Natural Deduction System

Cont...

- If a formula β is derived / proved from a set of premises / hypotheses $\{ \alpha_1, \dots, \alpha_n \}$,
  - then one can write it as **from** $\alpha_1, \dots, \alpha_n$ **infer** β.
- In natural deductive system,
  - a theorem to be proved should have a form **from** $\alpha 1, \dots, \alpha n$ **infer** β.
- Theorem **infer** β means that
  - there are no premises and β is true under all interpretations i.e., β is a tautology or valid.

Natural Deduction System

Cont..

- If we assume that $\alpha \rightarrow \beta$ is a premise, then we conclude that β is proved if α is given i.e.,
  - if 'from α infer β' is a theorem then $\alpha \rightarrow \beta$ is concluded.
  - The converse of this is also true.

Deduction Theorem: Infer $(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n \rightarrow \beta)$ is a theorem of natural deductive system if and only if

from $\alpha_1, \alpha_2, \dots, \alpha_n$ infer β is a theorem.

Useful tips: To prove a formula $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n \rightarrow \beta$, it is sufficient to prove a theorem **from $\alpha_1, \alpha_2, \dots, \alpha_n$ infer** β.

Natural Deduction System

Example

Example1: Prove that $P\wedge(QVR)$ follows from $P\wedge Q$

Solution: This problem is restated in natural deductive system as **"from $P \wedge Q$ infer $P \wedge (Q \vee R)$"**. The formal proof is given as follows:

{Theorem}	from $P \wedge Q$ infer $P \wedge (Q \vee R)$	
{ premise}	$P \wedge Q$	(1)
{ E-$\wedge$, (1)}	P	(2)
{ E-$\wedge$, (1)}	Q	(3)
{ I-V , (3) }	$Q \vee R$	(4)
{ I-$\wedge$, (2, 4)}	**$P \wedge (Q \vee R)$**	**Conclusion**

2)Write notes on certainty factor

The certainty-factor model was one of the most popular model for the representation and manipulation of uncertain knowledge in the early (1980s) Rule-based expert systems.

The **Certainty Factor (CF)** is a numeric value which tells us about how likely an event or a statement is supposed to be true. It is somewhat similar to what we define in probability, but the difference in it is that an agent after finding the probability of any event to occur cannot decide what to do. Based on the probability and other knowledge that the agent has, this **certainty factor** is decided through which the agent can decide whether to declare the statement true or false.

The value of the **Certainty factor** lies between **-1.0 to +1.0**, where the negative 1.0 value suggests that the statement can never be true in any situation, and the positive 1.0 value defines that the statement can never be false. The value of the **Certainty factor** after analyzing any situation will either be a positive or a negative value lying between this range. The value 0 suggests that the agent has no information about the event or the situation.