Knowledge Inference

Knowledge based agent

A knowledge based agent includes :

- A knowledge base
- Inference system

A knowledge base is a set of representations of facts of the world.

The agent operates as follows.

- It **tells** the knowledge base what it perceives.
- It **asks** the knowledge base what action it should perform.
- It **performs** the chosen action.

The knowledge base stores facts about the world. The inference engine/system applied logical rules to the knowledge base and deduced new knowledge.

Knowledge Inference

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts.

Inference engine commonly proceeds in two modes, which are:

- Forward chaining :Forward chaining starts with the known facts and asserts new facts.
- Backward chaining :Backward chaining starts with goals, and works backward to determine what facts must be asserted so that the goals can be achieved.

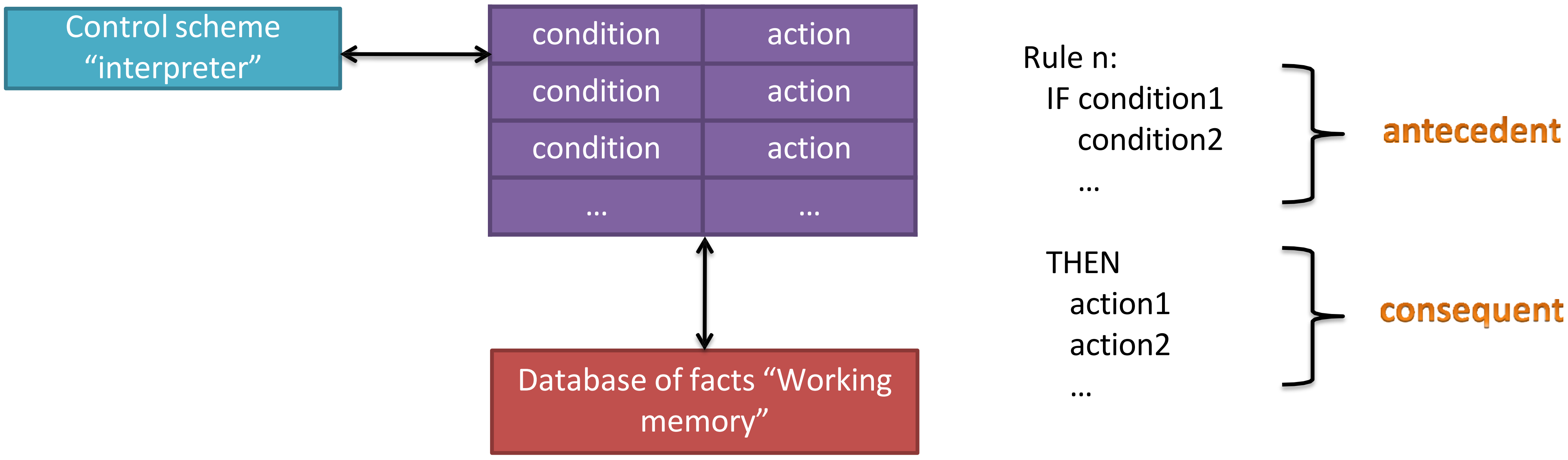
Knowledge Inference

- **Horn Clause and Definite clause:**
- Horn clause and definite clause are the forms of sentences, which enables knowledge base to use a more efficient inference algorithm.
- Logical inference algorithms use forward and backward chaining approaches, which require KB in the form of the **first-order definite clause**.
- **Definite clause:** A clause which is a disjunction of literals with **exactly one positive literal** is known as a definite clause or **strict horn clause**.
- **Horn clause:** A clause which is a disjunction of literals with **at most one positive literal** is known as horn clause. Hence all the definite clauses are horn clauses.

➢ **Example: (¬ p V ¬ q V k)**. It has only one positive literal k.

- It is equivalent to $p \land q \rightarrow k$.

Rule-Based Systems



- When one part of the IF portion matches a fact in working memory, the antecedent is SATISFIED. When all antecedents are satisfied, the rule is TRIGGERED. When the consequent of a rule is performed, the rule is FIRED.

Forward Chaining

- Forward chaining is data-driven. It is also known as a **forward deduction or forward reasoning method** when using an inference engine.
- Forward chaining is a form of reasoning which start with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached.
- The Forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process repeats until the problem is solved.
- E.g. A, A->B, B. A is the starting point. A->B represents a fact. This fact is used to achieve a decision B.
- E.g. Tom is running (A), If a person is running, he will sweat (A->B), Therefore, Tom is sweating. (B)

Forward Chaining

- **Properties of Forward-Chaining:**

➢ It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.

➢ Forward chaining approach is also called as data-driven as we reach to the goal using available data.

➢ Forward chaining approach is commonly used in the expert system, business, planning, monitoring, control and interpretation application, automated inference engines.

Forward Chaining

- Example:

"As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen."

Prove that **"Robert is criminal."**

- ➢ To solve the above problem, first, we will convert all the above facts into first-order definite clauses, and then we will use a forward-chaining algorithm to reach the goal.

Forward Chaining

- Facts Conversion into FOL:

1. It is a crime for an American to sell weapons to hostile nations. (Let's say p, q, and r are variables)

   American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p) ...(1)

2. Country A has some missiles.

- **∃ p Owns(A, p) ∧ Missile(p)**. It can be written in two definite clauses by using Existential Instantiation, introducing new Constant T1.

  Owns(A, T1) ......(2)

  Missile(T1) ....(3)

3. All of the missiles were sold to country A by Robert.

   ∀ p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)

4. Missiles are weapons.

   Missile(p) → Weapons (p) ....(5)

Forward Chaining

5. Enemy of America is known as hostile.

   Enemy(p, America) →Hostile(p) ....(6)

6. Country A is an enemy of America.

   Enemy (A, America) ....(7)

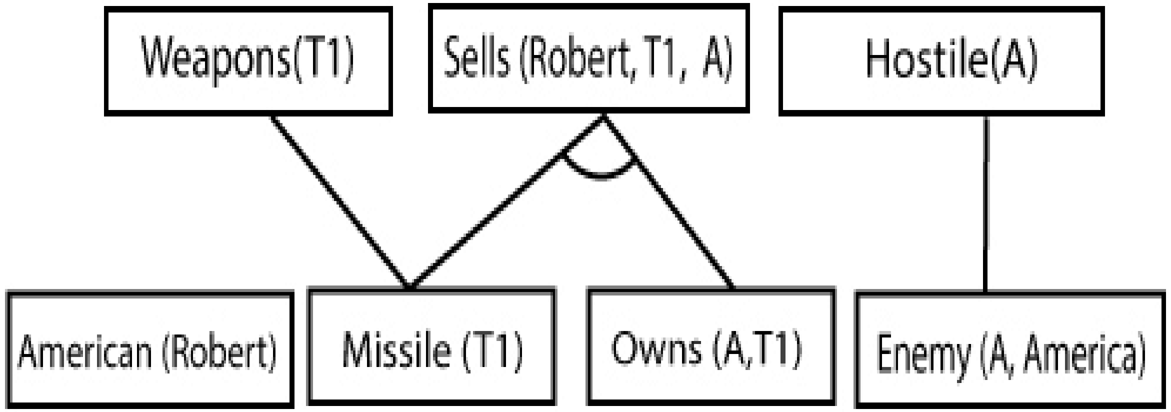
7. Robert is American

   American(Robert). ....(8)

Forward Chaining

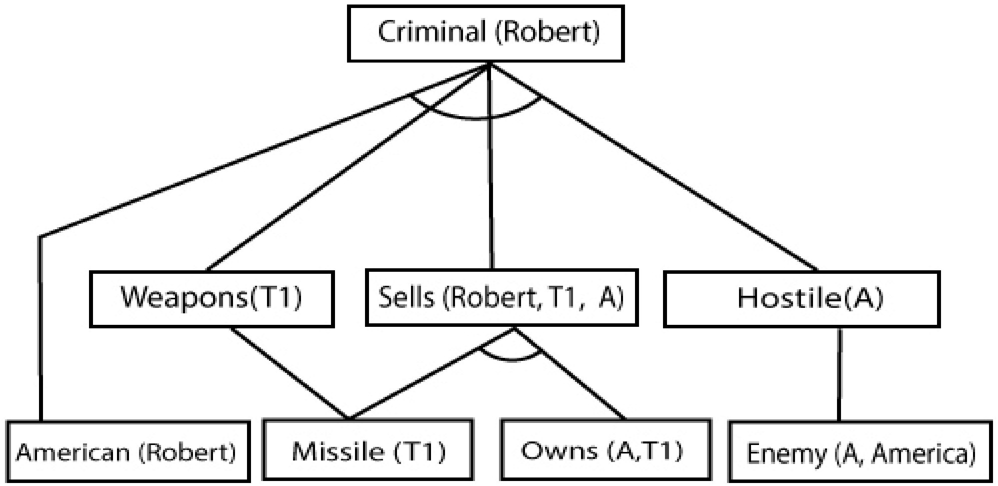
- **Step-1:**
- In the first step we will start with the known facts and will choose the sentences which do not have implications, such as:

  American(Robert), Enemy(A, America), Owns(A, T1), and Missile(T1).
- **Step-2:**
- At the second step, we will see those facts which infer from available facts and with satisfied premises.
- Rule-(1) does not satisfy premises, so it will not be added in the first iteration.
- Rule-(2) and (3) are already added.
- Rule-(4) satisfy with the substitution {p/T1}, **so Sells (Robert, T1, A)** is added, which infers from the conjunction of Rule (2) and (3).
- Rule-(6) is satisfied with the substitution(p/A), so Hostile(A) is added and which infers from Rule-(7).


Weapons(T1)
Sells (Robert, T1, A)
Hostile(A)
American (Robert)
Missile (T1)
Owns (A,T1)
Enemy (A, America)

Forward Chaining

- **Step-3:** At step-3, as we can check Rule-(1) is satisfied with the substitution **{p/Robert, q/T1, r/A}, so we can add Criminal(Robert)** which infers all the available facts. And hence we reached our goal statement.



- Hence it is proved that Robert is Criminal using forward chaining approach.

Forward Chaining

- **Advantages**
  - It can be used to draw multiple conclusions.
  - It provides a good basis for arriving at conclusions.
  - It's more flexible than backward chaining because it does not have a limitation on the data derived from it.
- **Disadvantages**
  - The process of forward chaining may be time-consuming. It may take a lot of time to eliminate and synchronize available data.
  - Unlike backward chaining, the explanation of facts or observations for this type of chaining is not very clear. The former uses a goal-driven method that arrives at conclusions efficiently.

Forward Chaining

There are two main types of control schemes that are applied to rule-based systems.

Z1 If ?x has hair
        Then ?x is a mammal
Z2 If ?x gives milk
        Then ?x is a mammal
Z3 If ?x has feathers
        Then ?x is a bird
Z6 If ?x is a mammal
                ?x has pointed teeth
                ?x has claws
                ?x has forward-pointing eyes
        Then ?x is a carnivore
Z8 If ?x is a mammal
                ?x chews cud
        Then ?x is an ungulate
Z11 If ?x is an ungulate
                ?x has long legs
                ?x has long neck
                ?x has tawny color
                ?x has dark spots
        Then ?x is a giraffe

Database

F1) Stretch has hair
F2) Stretch chews cud
F3) Stretch has long legs
F4) Stretch has a long neck
F5) Stretch has tawny color
F6) Stretch has dark spots

Backward Chaining

- Backward-chaining is also known as a **backward deduction or backward reasoning** method when using an inference engine.
- A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.
- Backward chaining can be used in debugging, diagnostics, and prescription applications.
- Backward chaining algorithm is also used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.

Backward Chaining

- When a query q is asked
  - if a matching fact q' is known, return the unifier
  - for each rule whose consequent q' matches q
  - attempt to prove each premise of the rule by backward chaining
- (Some added complications in keeping track of the unifiers)
- (More complications help to avoid infinite loops)
- Two versions: find **any** solution, find **all** solutions
- Backward chaining is the basis for **logic programming**, e.g., Prolog

Backward Chaining

- **Properties of backward chaining:**
- In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.
- It is called a goal-driven approach, as a list of goals decides which rules are selected and used.
- The modus ponens inference rule is used as the basis for the backward chaining process. This rule states that if both the conditional statement (p->q) and the antecedent (p) are true, then we can infer the subsequent (q).

Backward Chaining

- Reasons BACKWARD from goal through rules to facts
- Use modus ponens
  - Start at goals
  - Match goals to consequents or facts
- Repeat until
  - All sub goals are proven or
  - At least one sub goal cannot be proven
- Backward chaining is “goal driven”

Backward Chaining

- Backward chaining can be explained in the following sequence.
- B
- A->B
- ➢ A
- B is the goal or endpoint, that is used as the starting point for backward tracking. A is the initial state. A->B is a fact that must be asserted to arrive at the endpoint B.

Example: Tom is sweating (B).

- If a person is running, he will sweat (A->B).
- ➢ Tom is running (A).

Backward Chaining

- **Advantages**
  - The result is already known, which makes it easy to deduce inferences.
  - It's a quicker method of reasoning than forward chaining because the endpoint is available.
  - In this type of chaining, correct solutions can be derived effectively if pre-determined rules are met by the inference engine.
- **Disadvantages**
  - The process of reasoning can only start if the endpoint is known.
  - It doesn't deduce multiple solutions or answers.
  - It only derives data that is needed, which makes it less flexible than forward chaining.

Backward Chaining

- The backward-chaining method mostly used a **depth-first search** strategy for proof.
- **Example**:
- In backward-chaining, we will use the same above example, and will rewrite all the rules.
- **American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p) ...(1)**
Owns(A, T1) ....(2)
- **Missile(T1)**
- **?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)**
- **Missile(p) → Weapons (p) ....(5)**
- **Enemy(p, America) →Hostile(p) ....(6)**
- **Enemy (A, America) ....(7)**
- **American(Robert). ....(8)**

Backward Chaining

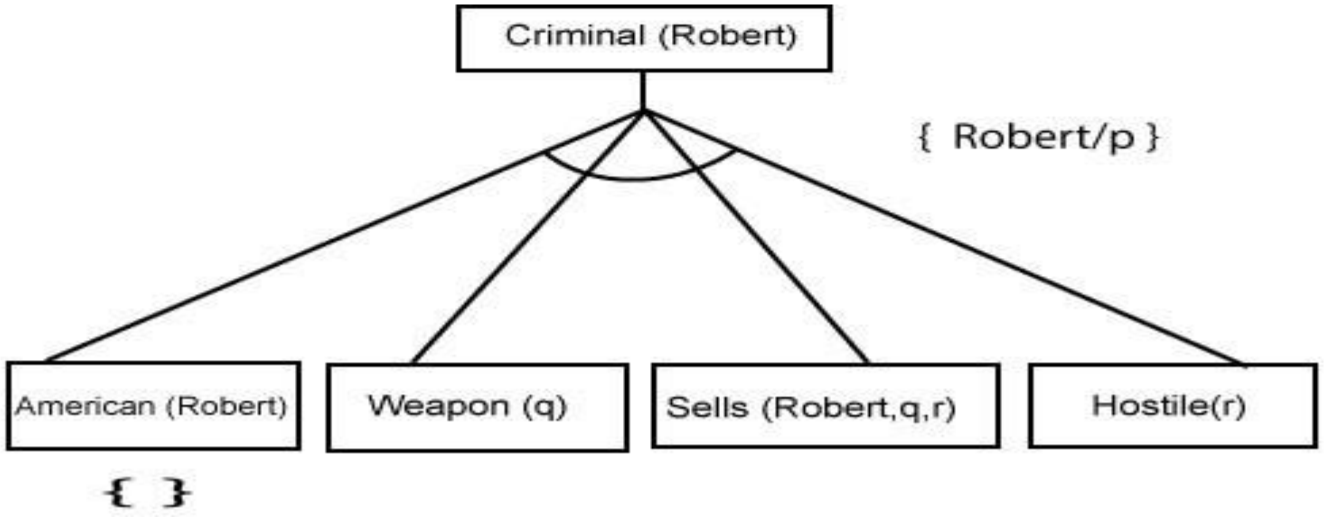
- Backward-Chaining proof:
- In Backward chaining, we will start with our goal predicate, which is **Criminal(Robert)**, and then infer further rules.
- **Step-1:**

➢ At the first step, we will take the goal fact. And from the goal fact, we will infer other facts, and at last, we will prove those facts true. So our goal fact is "Robert is Criminal," so following is the predicate of it.

- **Step-2:**

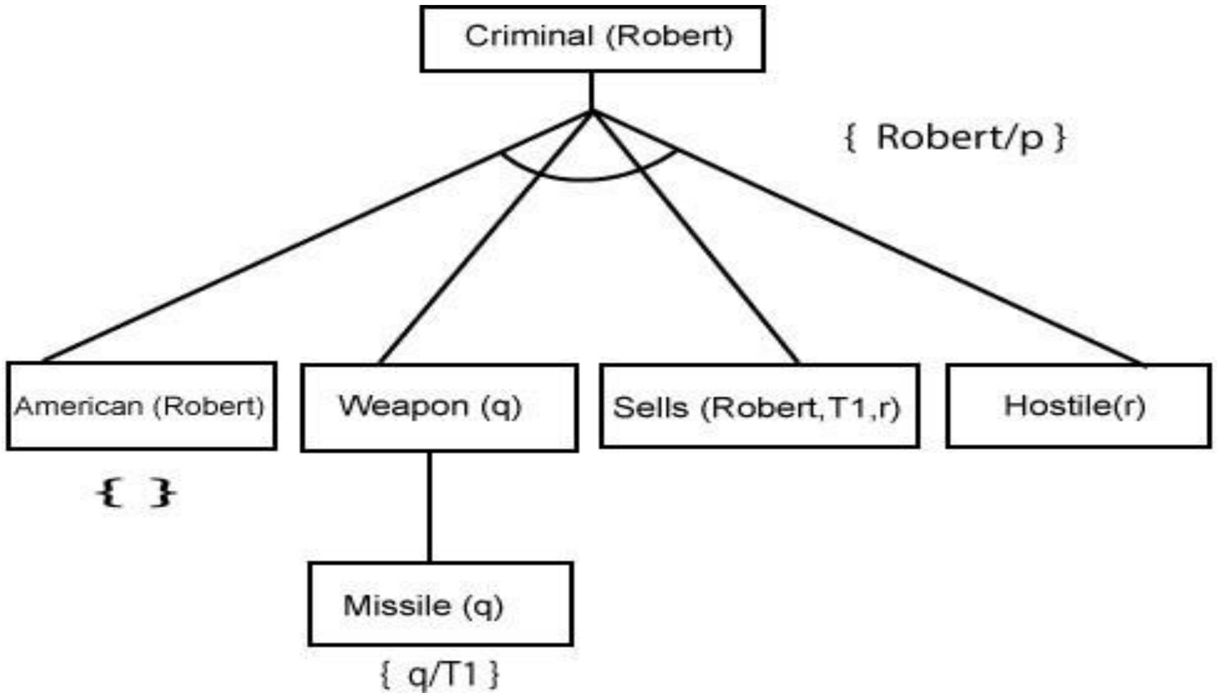
➢ At the second step, we will infer other facts form goal fact which satisfies the rules. So as we can see in Rule-1, the goal predicate Criminal (Robert) is present with substitution {Robert/P}. So we will add all the conjunctive facts below the first level and will replace p with Robert.

Backward Chaining



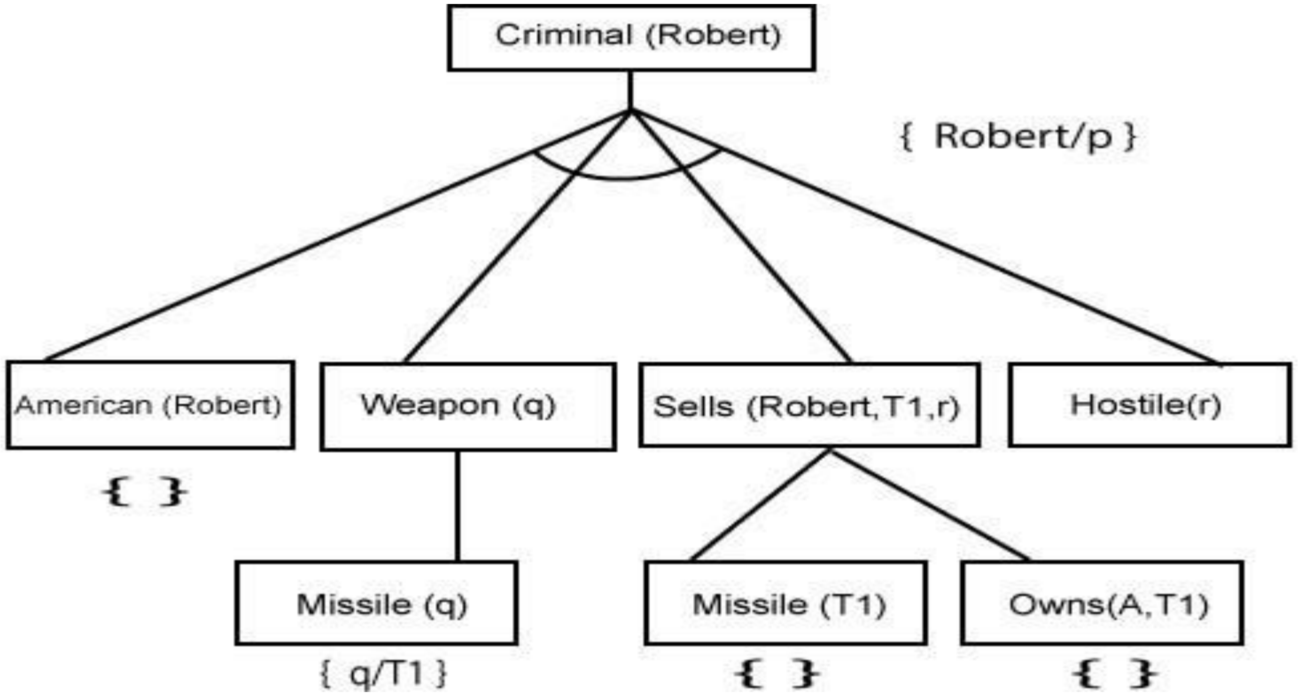
- **Step-3:**t At step-3, we will extract further fact Missile(q) which infer from Weapon(q), as it satisfies Rule-(5). Weapon (q) is also true with the substitution of a constant T1 at q.

Backward Chaining



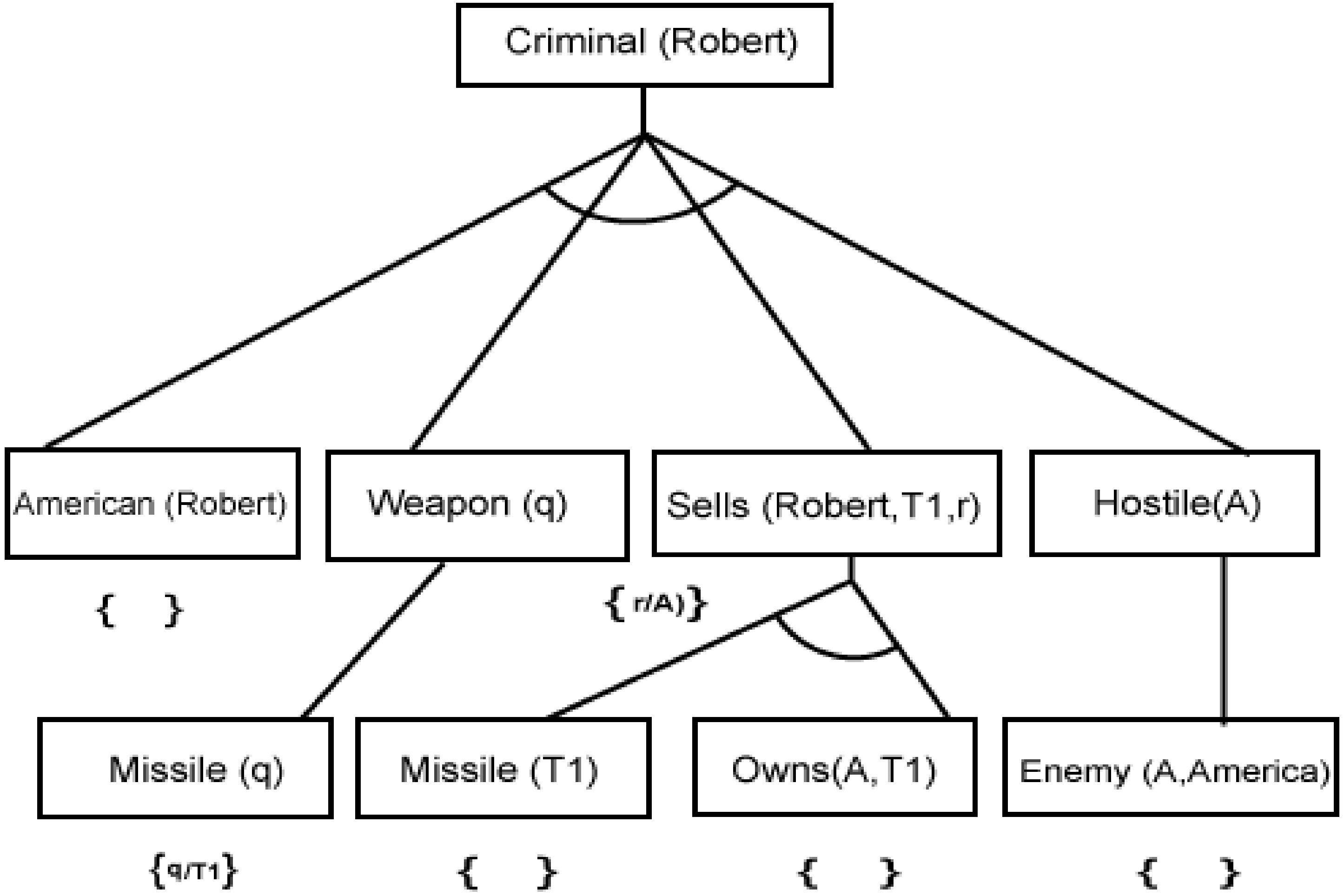
- **Step-4:**
- At step-4, we can infer facts Missile(T1) and Owns(A, T1) form Sells(Robert, T1, r) which satisfies the **Rule- 4**, with the substitution of A in place of r. So these two statements are proved here.

Backward Chaining



- **Step-5:**
- At step-5, we can infer the fact **Enemy(A, America)** from **Hostile(A)** which satisfies Rule- 6. And hence all the statements are proved true using backward chaining.

Backward Chaining



Forward Chaining vs. Backward Chaining

Forward Chaining	Backward Chaining
It starts from known facts and applies inference rule to extract more data unit it reaches to the goal.	It starts from the goal and works backward through inference rules to find the required facts that support the goal.
It is a top down approach.	It is a bottom-up approach.
It is known as data-driven inference technique as we reach to the goal using the available data.	It is known as goal-driven technique as we start from the goal and divide into sub-goal to extract the facts.
It applies a breadth-first search strategy.	It applies a depth-first search strategy
It tests for all the available rules	It tests only for few required rules.
It is suitable for the planning, monitoring, control, and interpretation application.	It is suitable for diagnostic, prescription, and debugging application.
It can generate an infinite number of possible conclusions.	It generates a finite number of possible conclusions.

Reasoning System

- The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs.
- It is a general process of thinking rationally, to find valid conclusions.

➢ Human reasoning capability are divided into three areas:

1. Mathematical reasoning
2. Logical reasoning
3. Non-logical reasoning

➢ These 3 areas of reasoning are in every human being but the ability level depends on education, environment and genetics.

Reasoning in AI

- To help machines perform human-level functions, AI and its various subfields like machine learning, deep learning, natural language processing rely on reasoning and knowledge representation.
- Reasoning in artificial intelligence helps machines think rationally and perform functions like humans.
- It is an important area of research in AI that helps machines in problem-solving, deriving logical solutions, and making predictions based on the available information, knowledge, facts, and data.

Information / Knowledge Base

- **Properties of Information**:
  - **Complete**: All the facts that are necessary to solve a problem are present in the system. E.g., Question Answering, or Proof.
  - **Consistent**: There is no contradiction in the available information. E.g., we should not be able to derive both P and ¬P.
  - **Monotonous/ Monotonically Growing**: An addition of information does not make existing information false, or inconsistent.

- When the information is not complete, or inconsistent, we need **Non-Monotonic Reasoning Systems**.

Uncertainty in AI

- Uncertainty plays a significant role in artificial intelligence.
- It refers to the lack of complete information or the presence of randomness and variability in data or in the outcomes of AI models.
- Dealing with uncertainty is crucial in AI for making informed decisions, handling noisy data, and building robust and reliable systems.

Applications:

1. Medical Diagnosis: In healthcare, uncertainty is prevalent due to variations in patient data and diagnostic tests.
2. Autonomous Vehicles: Self-driving cars encounter uncertainty in real-time situations, such as the behavior of other vehicles and road conditions.

Types of Reasoning

- **Types of Reasoning**

1. Monotonic Reasoning
2. Non Monotonic Reasoning
3. Common Sense Reasoning
4. Deductive reasoning
5. Inductive reasoning
6. Abductive reasoning

Monotonic Reasoning

- In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.
- To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.
- Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.
- Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.
- Any theorem proving is an example of monotonic reasoning.

Monotonic Reasoning

- **Example:**
- ➢ **Earth revolves around the Sun.**
- It is a true fact, and it cannot be changed even if we add another sentence in knowledge base like, "The moon revolves around the earth" Or "Earth is not round," etc.

Monotonous Information

- Conventional reasoning system works with information that is:
  - Complete
  - Consistent
  - Monotonous.

- When do you say that the information is monotonous?
  - If a new fact gets added to the already existing information, and still, all the information remains the same, it does not change, consistency still remains across all the facts, and no fact has to be retracted, then, this information is said to be monotonous.

Non Monotonic Reasoning

- In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.
- Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.
- Non-monotonic reasoning deals with incomplete and uncertain models.
- "Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

Non Monotonic Reasoning

- **Example:** Let suppose the knowledge base contains the following knowledge:

➢ **Birds can fly**

➢ **Penguins cannot fly**

➢ **Pitty is a bird**

- So from the above sentences, we can conclude that **Pitty can fly**.
- However, if we add one another sentence into knowledge base "**Pitty is a penguin**", which concludes "**Pitty cannot fly**", so it invalidates the above conclusion.

Non Monotonous Information

- ABC Murder Story.
  - A, B, C → suspects in a murder case.
  - A has an entry in the register of a respectable hotel at Albany, not at the place of the crime.
  - B's relative has testified that B was visiting him at Brooklyn at the time.
  - C claims to have been watching a ski meet at Catskills.
- So, our belief, now, is that,
  1. A did not commit the crime
  2. B did not commit the crime
  3. A or B or C committed the crime
- Now, C shows evidence that he was shown on TV, in the sidelines, at the ski meet. Now we have a new belief that,
  4. C did not commit the crime.
- Inconsistency!
- Reject the belief with the weakest evidence.
- Addition of a new fact 4, has made the existing information inconsistent. That's why our information is non-monotonous.

Non Monotonic Reasoning

- The definite clause logic is **monotonic** in the sense that anything that could be concluded before a clause is added can still be concluded after it is added; adding knowledge does not reduce the set of propositions that can be derived.
- A logic is **non-monotonic** if some conclusions can be invalidated by adding more knowledge. The logic of definite clauses with negation as failure is non-monotonic. Non-monotonic reasoning is useful for representing defaults. A **default** is a rule that can be used unless it overridden by an exception.

Non Monotonic Reasoning

- **Example**
- All balls bounce
- Football is a ball
- Does football bounce (Of course?)
- What about a football with no air filled in? (Conclusions change with new facts)

Non Monotonic Reasoning

- **Handling Exceptions** - We know

➢ Ostrich(x) → Abnormal(x)

➢ Ostrich is not a normal bird.

- We conclude

➢ Ostrich(x) → Bird(x) ^ ¬Flies(x) (We make all exceptions this way)

Non-monotonic Reasoning

- Non-monotonic reasoning can be describe using default reasoning.
- Default reasoning- The conclusions are drawn based on what is most likely to be true.
- There are 2 approaches:

➢ Non monotonic logic

➢ Default logic

- **Non monotonic logic**: The truth of a proposition may change when new information are added and a logic may be build to allows the statement to be retraced.
- Non monotonic logic is predicate logic with one extension called **model operator (M)** which means "consistent with everything we know".
- The purpose of M is to allow consistency.

Non-monotonic Reasoning

- NMR is useful for representing the **defaults**.
- Default is a rule that can be used unless it is overridden by exception.
- Default is a value that can be changed if exception occur else it remains the same as its original value.
- NMR is reasoning in which rules of inference are extended to make it possible to reason with incomplete information.

Default logic

- Logic base system is not able to handle non monotonous information

- Standard Logic → true or false
- Consider a rule "Birds can fly."
- Default Logic can formalise rules without explicitly mentioning all their exceptions.
- The rule can be formalized by the following 'default':

Precondition Justification

$$D = \left\{ \frac{Bird\ (X) : Flies\ (X)}{Flies\ (X)} \right\}$$

Conclusion

This rule says that, "If X is a bird, and it can be assumed that it flies, then we can conclude that it flies."

- Consider the following is the knowledge that we have:

$$W = \{Bird(Pigeon), Flies(Pigeon), Bird(Penguin), \neg Flies(Penguine), Flies(Bee)\}$$

- Consider X as pigeon. We can conclude that a pigeon flies.
- Now, consider X is a penguin. Flies(penguin) is not consistent with what is known. That is why, you cannot conclude that penguin can fly.
- Now consider that X is a Bee.
- Default rule allows a delivery from the precondition to the justification and not from the justification to the precondition. Since for a bee the precondition is not satisfied, you cannot make a conclusion that the bee flies.

Default logic

- A default theory is a pair <W, D>, where W is a set of logical formulas called the background theory, that formalizes the facts that are known for sure and D is a set of default rules, also called 'Defaults' each one being of the form:

$$\frac{Prerequisite : Justification_1, \dots, Justification_n}{Conclusion}$$

- It says that, if we believe that prerequisite is true, and each of the justifications is consistent with our current beliefs, we are led to believe that conclusion is true.

- For the ABC murder story, we can have a default like the following:

$$D = \left\{ \frac{Accused\ (X): Innocent\ (X)}{Innocent\ (X)} \right\}$$

- Which says that if X is the accused and it is consistent to assume that he is innocent, then we can conclude that he is innocent.

Non Monotonic Inference

- **Non-monotonic reasoning**:
  - An approach to reason with incomplete information.
  - Add axioms/ rules to form an extended knowledge base.
  - Extended knowledge base is obtained using the existing knowledge and our new information obtained using non-monotonic reasoning.
  - An inference based on lack of knowledge is called Non-Monotonic Inference.
- Consider the following Knowledge Base:
  - Pooja likes ice cream
  - Pallavi likes ice cream.
- Does Ravina like ice cream?
  - The database does not have that information
  - The answer could be, false, Ravina does not like ice cream. → Non-Monotonic Inference

Closed World Assumption

- **Closed World Assumption** is an approach for dealing with incompleteness by assuming that, anything that is not contained within the knowledge base is false.
- Let us consider another example where the knowledge base consists of 5 cities, A, B, C, D and E, and the following information is available:
  - City A is connected to city B
  - City B is connected to city C
  - City C is connected to city D
  - City D is connected to city E
- Then the extended knowledge base will consists of statements that mention which is not connected to which city.
- Is city A connected to city E? → No!
- Our extended database says that, city A is not connected to city E.
- This would be the Non-Monotonic Inference.

Limitations of Closed World Assumption

1. Knowledge is not realistically "closable".
   - For example in the ABC Murder story W
     - A or B or C committed the crime
     - A did not commit the crime
     - B did not commit the crime
   - This was our closed world. C's proof was not present in the knowledge base.
2. Syntax problem: Conjunction can be handled by CWA. But not Disjunction.

Knowledge Base: single(Jim) ∨ single(Pam)	
Is Jim single? → The answer cannot be yes	Is Pam single? → The answer cannot be yes
¬single(Jim)	¬single(Pam)

   - So the extended database would look like this
     - Single(Jim) ∨ Single(Pam)
     - ¬ single(Jim)
     - ¬ single(Pam)
   - This extended data base is inconsistent!
   - This example shows that **Disjunctions cannot be handled by the Closed World Assumption.**

Generalized Closed World Assumption

- The Generalised CWA allows addition of statements only if it does not make the existing knowledge base inconsistent.

Knowledge Base: single(Jim) v single(Pam)	
Is Jim single? →The answer cannot be yes	Is Pam single? →[illegible]wer cannot be yes
¬single(Jim)	¬s[illegible]m)

- So the extended Knowledge base using GCWA looks like this:
    - Single(Jim) V Single(Pam)
    - ¬single(Jim)
- This keeps the knowledge base consistent. But not necessarily complete.

Common sense Reasoning

- Common sense reasoning is an informal form of reasoning, which can be gained through experiences.
- Common Sense reasoning simulates the human ability to make presumptions about events which occurs on every day.
- It relies on good judgment rather than exact logic and operates on **heuristic knowledge** and **heuristic rules**.
- **Example:**

➢ **One person can be at one place at a time.**

➢ **If I put my hand in a fire, then it will burn.**

- The above two statements are the examples of common sense reasoning which a human mind can easily understand and assume.
- It is mainly used in computer vision and robotic manipulation.

Deductive Reasoning

- **Deductive reasoning is deducing new information from logically related known information. It is the form of valid reasoning, which means the argument's conclusion must be true when the premises are true.**
- It is sometimes referred to as top-down reasoning, and contradictory to inductive reasoning.
- In deductive reasoning, the truth of the premises guarantees the truth of the conclusion.
- Deductive reasoning mostly starts from the general premises to the specific conclusion, which can be explained as below example.
- **Example:**

➢ **Premise-1: All the human eats veggies**

➢ **Premise-2: Suresh is human.**

- **Conclusion: Suresh eats veggies.**

Deductive Reasoning

- Deductive reasoning uses formal logic to produce logically certain results, using a set process, which involves the following steps:

1. Theory
2. Hypothesis
3. Patterns
4. Confirmation

Inductive Reasoning

- An opposite of deductive reasoning, inductive reasoning is a **bottom-up** logic that uses specific observations to reach a conclusion. Used in cases where there is a limited set of facts and data to arrive at a conclusion.
- <u>Inductive reasoning uses historical data to produce a generic rule whose conclusion is supported by the premises.</u>
- Like Deductive reasoning, inductive reasoning also follows a set of steps to perform reasoning, these are:

1. Observation
2. Pattern
3. Hypothesis
4. Theory

Inductive vs deductive reasoning

- Deductive and inductive reasoning are opposites — deduction applies a top-to-bottom (general to specific) approach to reasoning whereas induction applies a bottom-to-top (specific to general) approach.
- **Inductive Reasoning** – Specific to general

➢ Logically true

➢ May or may not be realistically true (But not definitely true)

Statement 1 : Mango is a Fruit (Specific Statement)

Statement 2 : The box is full of Fruits (Specific Statement)

Conclusion : The box is full of Mangoes (General Conclusion)

Inductive vs deductive reasoning

- **Deductive Reasoning** – General argument to specific conclusion

➢ Logically true

➢ Realistically true (Always true)

Statement 1 : All mangoes are fruits (General Statement)

Statement 2 : All fruits have seeds (General Statement)

Conclusion : Mangoes have seeds (Specific Conclusion)

- The conclusion that we arrive based on inductive reasoning is called as **conjecture.**

Inductive vs deductive reasoning

- **Conjecture** is a hypothesis that has not be proven. Just because we observe a pattern in many cases does not mean it's true for all cases.
- Conjecture must be prove for that particular case. To prove such conjecture, principal of mathematical induction is used.

- **Deductive reasoning: conclusion guaranteed**
- **Inductive reasoning: conclusion merely likely**

Inductive reasoning

- Inductive reasoning starts with a specific assumption, then it broadens in scope until it reaches a generalized conclusion. With inductive reasoning, the conclusion may be false even if the premises are true.

➢ **Generalized Inductive Reasoning Example**:

- There are a total of 20 apples and oranges in a basket. I pulled out four apples and one orange, therefore there are 10 apples and five oranges in the basket.

➢ **Predictive Inductive Reasoning Example**:

- Most baseball players become coaches. Leo is a baseball player, so he’ll become a coach.

Inductive vs deductive reasoning

Think about this real-world problem:

To estimate the population of a town in upcoming years, one of the town workers collected populations from past years and made this table:

Year	Population
1950	7,403
1960	7,958
1970	8,377
1980	8,775
1990	9,323
2000	9,794
2010	10,281

The town wants to estimate the population for 2015, 2018, and 2020. To do this, will you be using inductive reasoning or deductive reasoning?

- **Inductive reasoning** is looking for a pattern or looking for a trend and then generalizing.

Inductive vs deductive reasoning

- **Inductive Reasoning**: The first lipstick I pulled from my bag is red. The second lipstick I pulled from my bag is red. Therefore, all the lipsticks in my bag are red.
- ➢ **Deductive Reasoning**: The first lipstick I pulled from my bag is red. All lipsticks in my bag are red. Therefore, the second lipstick I pull from my bag will be red, too.
- **Inductive Reasoning**: My mother is Irish. She has blond hair. Therefore, everyone from Ireland has blond hair.
- ➢ **Deductive Reasoning**: My mother is Irish. Everyone from Ireland has blond hair. Therefore, my mother has blond hair.
- **Inductive Reasoning**: Maximilian is a shelter dog. He is happy. All shelter dogs are happy.
- ➢ **Deductive Reasoning**: Maximillian is a shelter dog. All shelter dogs are happy. Therefore, he is happy.

Abductive Reasoning

Abductive reasoning is a form of logical reasoning which starts with single or multiple observations then seeks to find the most likely explanation or conclusion for the observation.

Abductive reasoning is an extension of deductive reasoning, but in abductive reasoning, the premises do not guarantee the conclusion.

- Example:

Implication: Cricket ground is wet if it is raining

Axiom: Cricket ground is wet.

Conclusion It is raining.

Inductive Reasoning Vs. Deductive Reasoning

- Inductive and deductive reasoning are the two most important and commonly used reasoning techniques.

INDUCTIVE REASONING	DEDUCTIVE REASONING
This type of reasoning reaches the conclusion through the process of generalization using facts and data.	Type of valid reasoning, where new information or conclusion is deduced using known facts and information.
It follows a bottom-up approach and starts from the conclusion.	It follows a top-down approach and starts from the premises.
The conclusion is not guaranteed to be true if the premises are true.	Here the conclusion is true if the premises are true.
It is fast and easy, as it requires evidence instead of facts.	Compared to inductive reasoning, it is difficult to use as it requires true facts.

Probabilistic Reasoning

- Till now, we have learned knowledge representation using first-order logic and propositional logic with certainty, which means we were sure about the predicates.
- With this knowledge representation, we might write A→B, which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called **uncertainty.**
- So to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or **probabilistic reasoning**.

Probabilistic Reasoning

- Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.
- We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.
- In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behaviour of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Probabilistic Reasoning

- **Need of probabilistic reasoning in AI:**

➢ When there are unpredictable outcomes.

➢ When specifications or possibilities of predicates becomes too large to handle.

➢ When an unknown error occurs during an experiment.

➢ In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

1. **Bayes' rule**
2. **Bayesian Statistics**

Bayes' Theorem

- P(H | E)

Hypothesis Evidence

- Truthiness is depend on how many evidence supporting the hypothesis.
- P(Hi | E) – What is probability of hypothesis that evidence is available.
- P(E | Hi) – Probability of evidence being present for a particular ith hypothesis.
- P(Hi) – Probability for ith hypothesis to be true (Whatever the evidence available or not)
- Baye's theorem:

$$P(H_i \mid E) = P(E|H_i)*P(H_i) \,/\, \sum P(E|H_n)*P(H_n) \text{ where } n=1 \text{ to } k.$$

Baye's Theorem

BAYE'S THEOREM: Describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

↳ In Probability theory it relates the Conditional probability & marginal probabilities of two random events.

↳ Calculate $P(B|A)$ with knowledge of $P(A|B)$.

↳ $P(H|E) = \dfrac{\text{no. of time H and E}}{\text{no. of times E}}$

$P(H|E) = \dfrac{P(H \cap E)}{P(E)}$ } Prob. of H when E is true.

Baye's Theorem

BAYE'S THEOREM: Describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

↳ In Probability theory it relates the Conditional probability & marginal probabilities of two random events.

↳ $P(H|E) = \dfrac{\text{no. of time H and E}}{\text{no. of times E}}$

↳ Calculate $P(B|A)$ with knowledge of $P(A|B)$.

$P(H|E) = \dfrac{P(H \cap E)}{P(E)}$ } Prob. of H when E is true.

$P(A \cap B) = P(A|B) \cdot P(B)$ — (i)

$P(A \cap B) = P(B|A) \cdot P(A)$ — (ii)

} From (i) and (ii) L.H.S are equal.

↳ $P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$

So, $\boxed{P(B|A) = \dfrac{P(A|B) \cdot P(B)}{P(A)}}$ Baye's theorem formula.

Posterior (Prob. of A when B is true)

marginal Prob. (Prob. of evidence)

Likelihood (Prob. of evidence)

Prior Prob (Prob. of hypothesis)

Baye's Theorem

Baye's theorem Example 1:

Ques 1:- what is the probability that Person has disease dengue with neck pain?

↳ Given:-

→ 80% of time dengue causes neck pain. $P(a|b) = 0.8$

↳ $P(\text{dengue}) = 1/30{,}000$ $(P(b) \Rightarrow 1/30{,}000$

↳ $P(\text{neckpain}) = .02$ $P(a) = .02$

a = Proposistion that Person has neck pain

b = Person has dengue.

$P(b|a) = ?$

$$P(b|a) = \frac{P(a|b) \cdot P(b)}{P(a)}$$

$$= \frac{0.8 \cdot 1/30000}{0.02} = \boxed{0.00133}$$

Baye’s Theorem

- Application of Baye’s theorem in AI
- ➢ Next step is calculated based on prior step
  - – Robot
  - – Automatic Machine
- ➢ Forecasting
  - – Weather

Probabilistic Reasoning

- As probabilistic reasoning uses probability and related terms, so before understanding probabilistic reasoning, let's understand some common terms:
- **Probability:** Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.
- $0 \leq P(A) \leq 1$, where $P(A)$ is the probability of an event A.
- $P(A) = 0$, indicates total uncertainty in an event A.
- $P(A) = 1$, indicates total certainty in an event A.

Probabilistic Reasoning

- We can find the probability of an uncertain event by using the below formula.

$$\text{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

- P(¬A) = probability of a not happening event.
- P(¬A) + P(A) = 1.

- **Event:** Each possible outcome of a variable is called an event.
- **Sample space:** The collection of all possible events is called sample space.
- **Random variables:** Random variables are used to represent the events and objects in the real world.
- **Prior probability:** The prior probability of an event is probability computed before observing new information.
- **Posterior Probability:** The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Probabilistic Reasoning

- **Conditional probability:**
- Conditional probability is a probability of occurring an event when another event has already happened.
- Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

- **Where P($A \wedge B$)= Joint probability of a and B**

  P(B)= Marginal probability of B.

- If the probability of A is given and we need to find the probability of B, then it will be given as:

$$P(B|A) = \frac{P(A \wedge B)}{P(A)}$$

Joint Probability Distribution

- Because events are rarely isolated from other events, we may want to define a joint probability distribution, or $P(X_1, X_2, .., X_n)$.
- Each X_i is a vector of probabilities for values of variable X_i.
- The joint probability distribution is an n-dimensional array of combinations of probabilities.

	Wet	~Wet
Rain	**0.6**	**0.4**
~Rain	0.4	0.6

Probabilistic Reasoning

- **Example:**
- In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like mathematics?

➢ **Solution:**

- Let, A is an event that a student likes Mathematics
- B is an event that a student likes English.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

- **Hence, 57% are the students who like Mathematics.**

Bayes' Theorem

- Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.
- In probability theory, it relates the conditional probability and marginal probabilities of two random events.
- Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.
- It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).
- Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Bayes' Theorem

- Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:
- As from product rule we can write: $P(A \wedge B) = P(A|B)\,P(B)$
- Similarly, the probability of event B with known event A $P(A \wedge B) = P(B|A)\,P(A)$
- Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad \text{\_\_\_\_\_\_(a)}$$

- The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

Bayes' Theorem

- It shows the simple relationship between joint and conditional probabilities. Here,
- P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.
- P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.
- P(A) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(B) is called **marginal probability**, pure probability of an evidence.
- In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where $A_1, A_2, A_3, \ldots\ldots, A_n$ is a set of events.

Applying Bayes' rule

- Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\mathbf{P(cause|effect)} = \frac{P(effect|cause)\,P(cause)}{P(effect)}$$

Applying Bayes' rule

- **Example-1: Question: what is the probability that a patient has diseases meningitis with a stiff neck?**
- **Given Data:** A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

1. The Known probability that a patient has meningitis disease is 1/30,000.
2. The Known probability that a patient has a stiff neck is 2%.

- Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

➢ P(a|b) = 0.8

➢ P(b) = 1/30000

➢ P(a)= .02

$$\mathbf{P(b|a)} = \frac{P(a|b)P(b)}{P(a)} = \frac{0.8*(\frac{1}{30000})}{0.02} = 0.001333333.$$

- Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Example

- We wish to know probability that John has malaria, given that he has a slightly unusual symptom: a high fever.
- We have 4 kinds of information
  a) probability that a person has malaria regardless of symptoms
  b) probability that a person has the symptom of fever given that he has malaria
  c) probability that a person has symptom of fever, given that he does NOT have malaria
  d) John has high fever
- H = John has malaria
- E = John has a high fever

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Suppose P(H) = 0.0001, P(E|H) = 0.75, P(E|~H) = 0.14

Then P(E) = 0.75 * 0.0001 + 0.14 * 0.9999 = 0.14006

and P(H|E) = (0.75 * 0.0001) / 0.14006 = 0.0005354

On the other hand, if John did not have a fever, his probability of having malaria would be

$$P(H|\sim E) = \frac{P(\sim E|H) * P(H)}{P(\sim E)} = \frac{(1-0.75)(0.0001)}{(1-0.14006)} = 0.000029$$

Which is much smaller.

Application of Bayes' theorem

Following are some applications of Bayes' theorem:

1. It is used to calculate the next step of the robot when the already executed step is given.
2. Bayes' theorem is helpful in weather forecasting.
3. It can solve the Monty Hall problem.

Bayesian Network/Belief Networks

- "A Bayesian network is a **probabilistic graphical model** which represents a set of variables and their conditional dependencies using a **directed acyclic graph(DAG)."**
- It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.
- Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

1. Directed Acyclic Graph
2. Table of conditional probabilities-probability of all the nodes with effect of parent node.

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

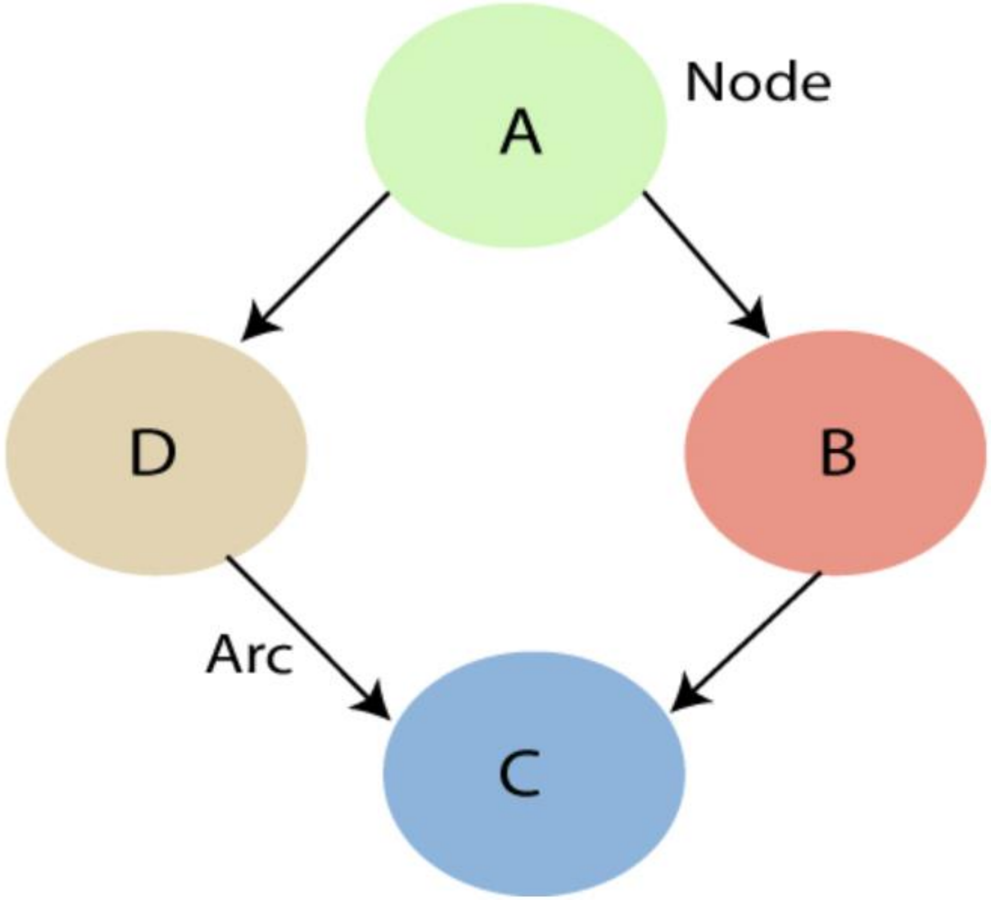
A Bayesian network graph is made up of nodes and Arcs (directed links), where:

•Each node corresponds to the random variables.

•Arc or directed arrows represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes.

•These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other.

- In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.
- If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.
- Node C is independent of node A.


Node
A
D
B
Arc
C

Bayesian Network

- DAG models the uncertainty of an event occurring based on conditional probability distribution of each random variable.
- Conditional probability table is used to represent this distribution of each variable in the network.
- **Joint Probability** is a measure of two events happening at the same time, i.e., p(A and B). The probability of the intersection of A and B may be written p(A ∩ B).

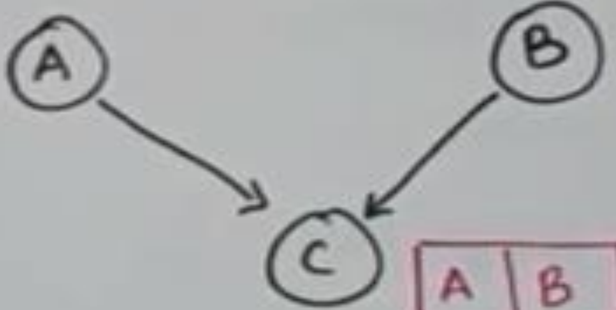
Conditional Probability of an event B is the probability that the event will occur given that an event A has already occurred.

p(B|A): probability of event B occurring, given that event A occurs.
If A and B are **dependent** events : *P (B|A) = P (A and B) / P (A)*
If A and B are **independent** events: *P(B|A) = P (B)*

Bayesian Belief Network

To propagate belief in Bayesian N/w, initial "Directed Acyclic Graph" is Converted into an undirected graph in which the arcs Can be used to transmit probabilities in dir^n of evidence.

T	.002
F	.998

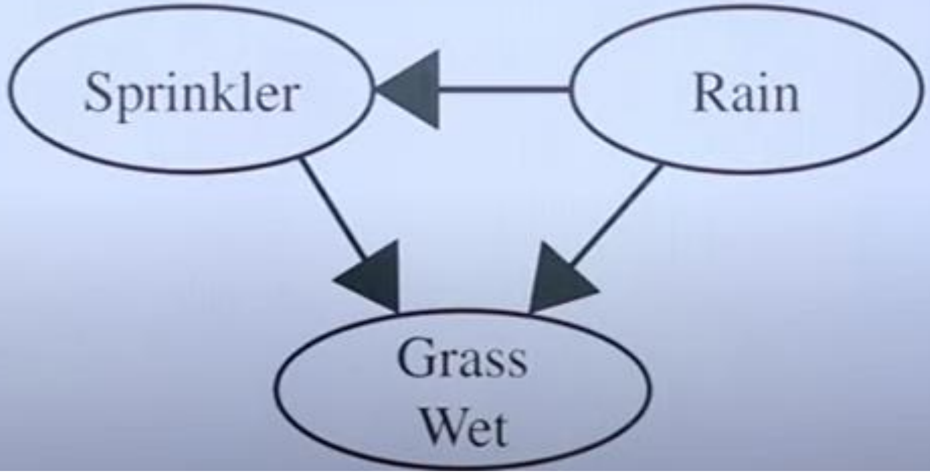


T	0.001
F	.999

A	B	P(C=T)	C=F
T	T	–	–
T	F	–	–
F	T	–	–
F	F	–	–

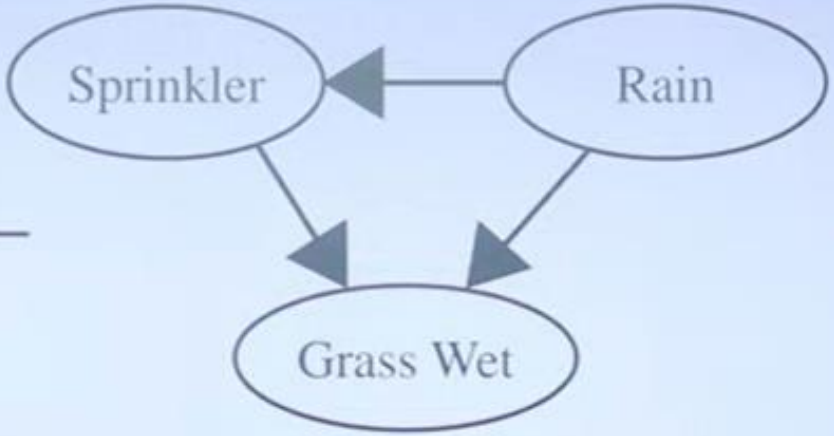
Bayesian network

It is probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).



Sprinkler

Rain	S
F	0.4
T	0.01



Rain

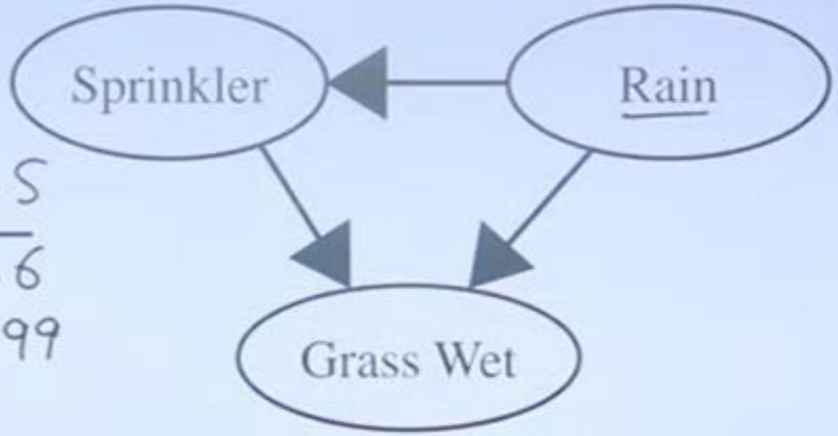
R
0.2

Grass Wet

Sprinkler	Rain	W
~S	~R	0.0
~S	R	0.8
S	~R	0.9
S	R	0.99

Sprinkler

Rain	S	~S
F	0.4	.6
T	0.01	.99



Rain

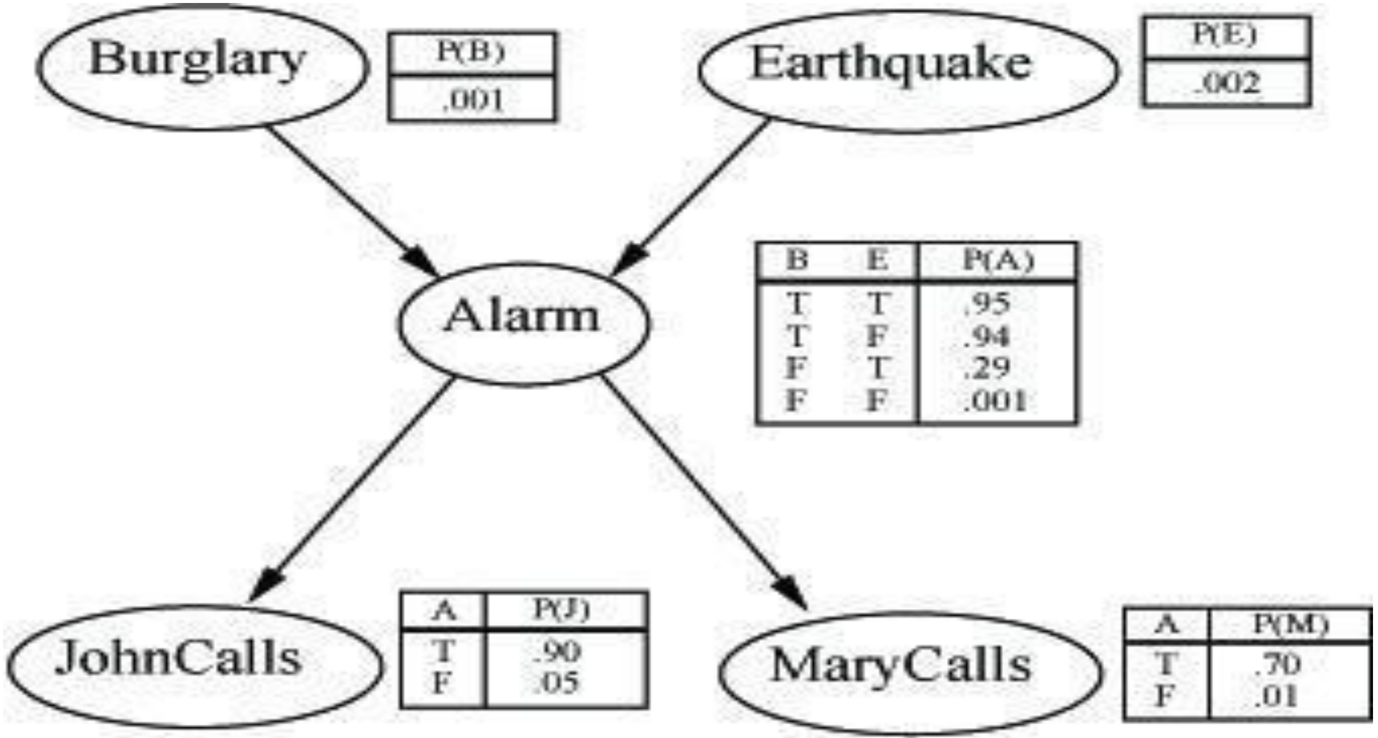
R	~R
0.2	.8

S R **Grass Wet**

Sprinkler	Rain	W	~W
~S	~R	0.0	1
~S	R	0.8	.2
S	~R	0.9	.1
S	R	0.99	0.01

Bayesian Network Example2

- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?
- Variables: Burglar, Earthquake, Alarm, JohnCalls, MaryCalls Network topology reflects “causal” knowledge:

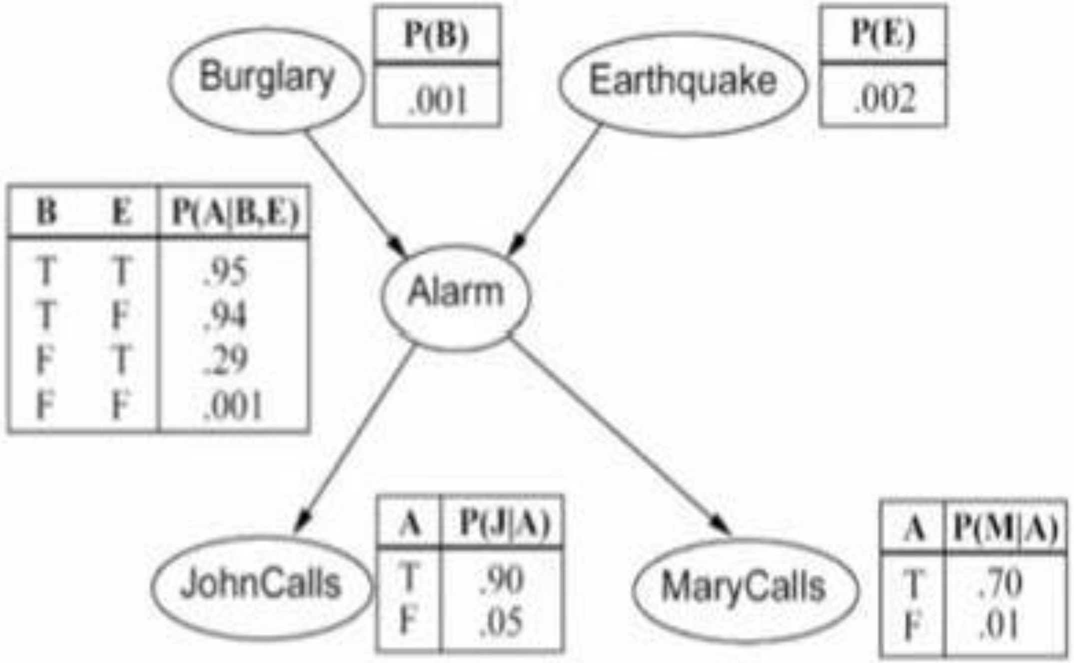


Bayesian Network Example2

- You have a new burglar alarm installed at home.
- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, John and Merry , who promised to call you at work when they hear the alarm.
- John always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Merry likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

Bayesian Network Example2

What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Merry call?



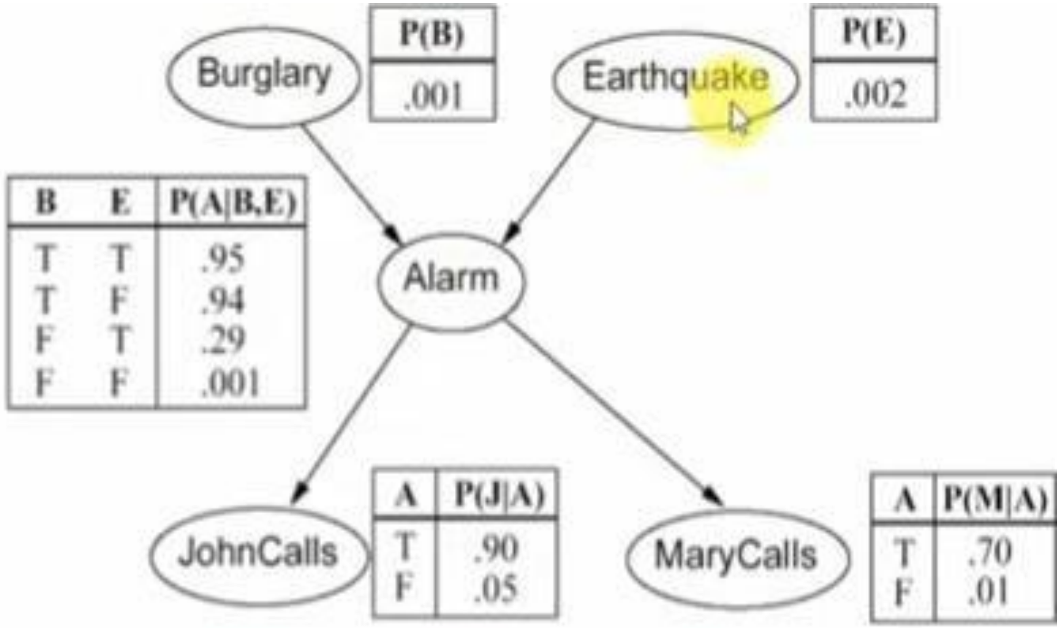
$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

Bayesian Network Example2

2. What is the probability that John call?



Solution:

$$P(j) = P(j \mid a)\, P(a) + P(j \mid \neg a)\, P(\neg a)$$

$$= P(j|a)\{P(a|b,e)*P(b,e)+P(a|\neg b,e)*P(\neg b,e)+P(a|b,\neg e)*P(b,\neg e)+P(a|\neg b,\neg e)*P(\neg b,\neg e)\}$$

$$+ P(j|\neg a)\{P(\neg a|b,e)*P(b,e)+P(\neg a|\neg b,e)* P(\neg b,e)+P(\neg a|b,\neg e)* P(b,\neg e)+P(\neg a|\neg b, \neg e)* P(\neg b, \neg e)\}$$

$$= 0.90 * 0.00252 + 0.05 * 0.9974 = 0.0521$$

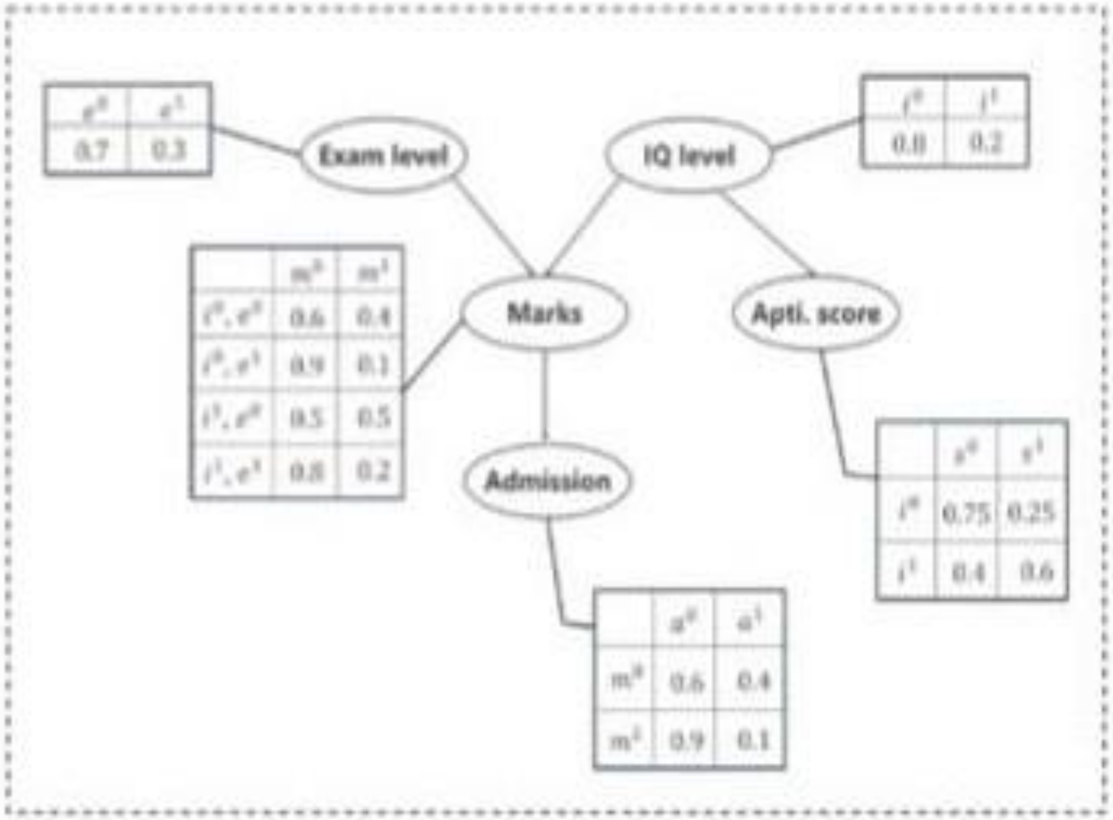
Bayesian Network Example

- There should not be a cycle.

Create a Bayesian Network that will model the marks (m) of a student on his examination.

The marks will depend on:

- ***Exam level*** *(e): (difficult, easy)*
- ***IQ*** *of the student (i): (high, low)*
- *Marks ->* ***admitted (a)*** *to a university*
- *The IQ ->* ***aptitude score (s)*** *of the student*

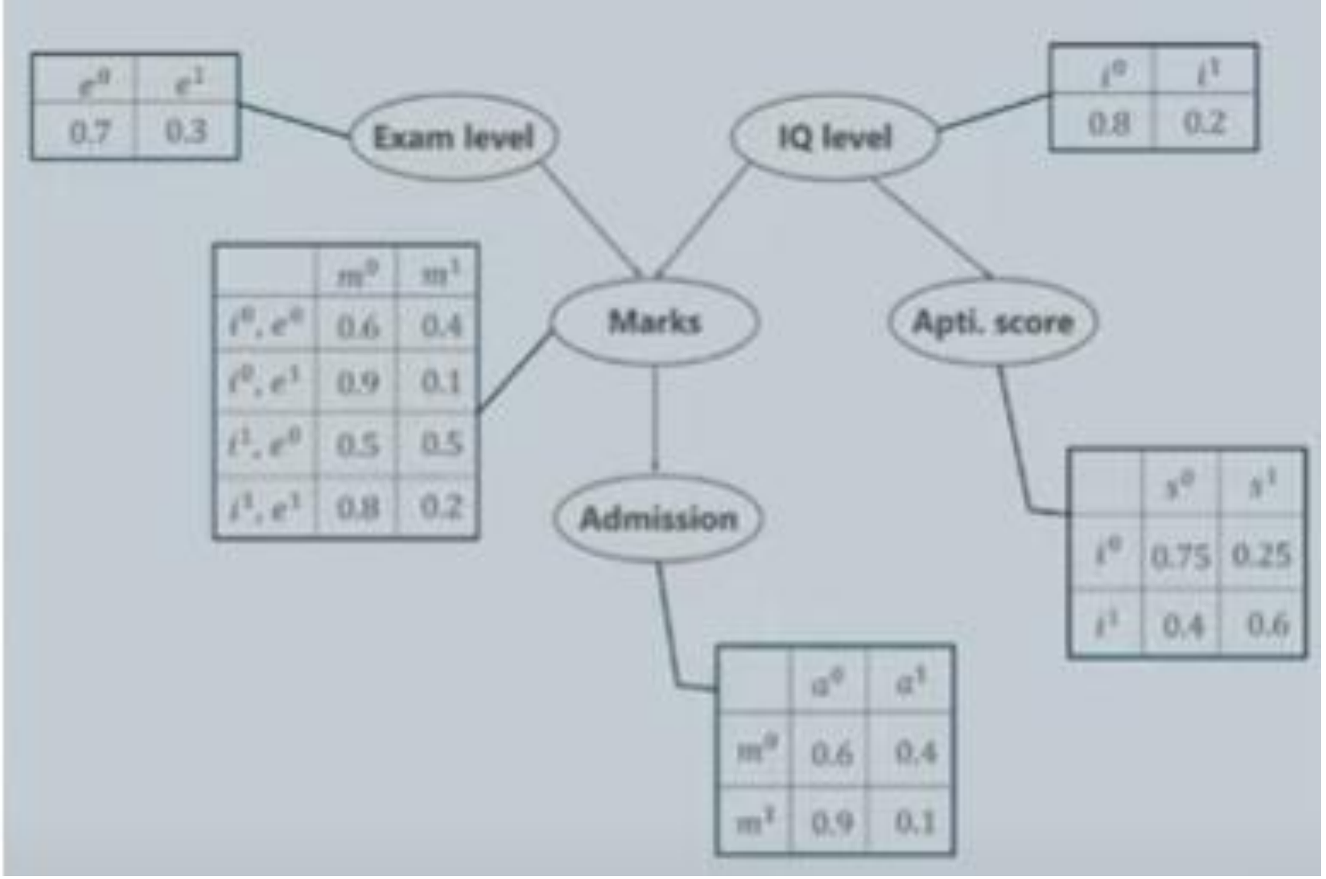


Bayesian Network Example

Factorizing Joint Probability Distribution:

p(a, m, i, e, s) = p(a | m) p(m | i, e) p(i) p(e) p(s | i)

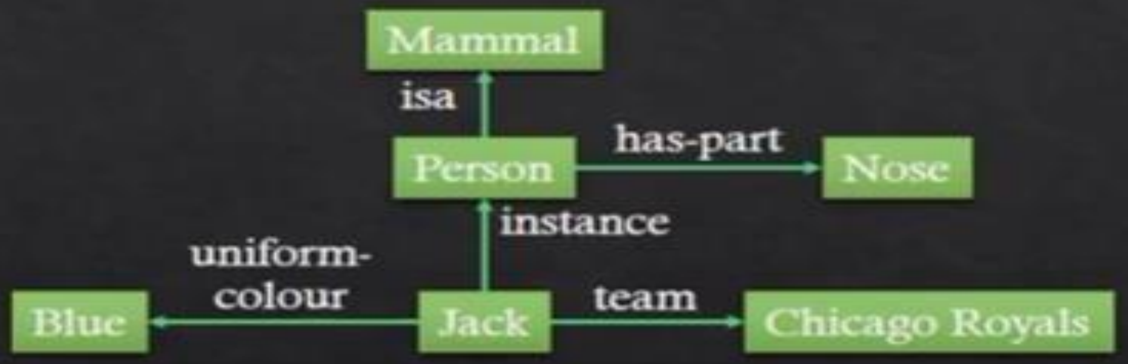
- p(a | m): CP of student admit -> marks
- p(m | i, d): CP of the student's marks -> (IQ & exam level)
- p(i): Probability -> IQ level
- p(d): Probability -> exam level
- p(s | i) CP of aptitude scores ->IQ level



Semantic Nets

- Semantic Nets – way of representing knowledge and are based on property inheritance.
- How to represent English statements using semantic nets?
- Represent knowledge with the help of property inheritance we take the help of structures called slot and filler structures.

- Slot and filler structures are the devices to support Property Inheritance using isa and instance links.



- **Arcs**: Represent relationships among the nodes
- **Nodes**: classes, or objects, or values of attributes of an object.
- A slot is an attribute-value pair and a filler is the actual value that the slot can take.

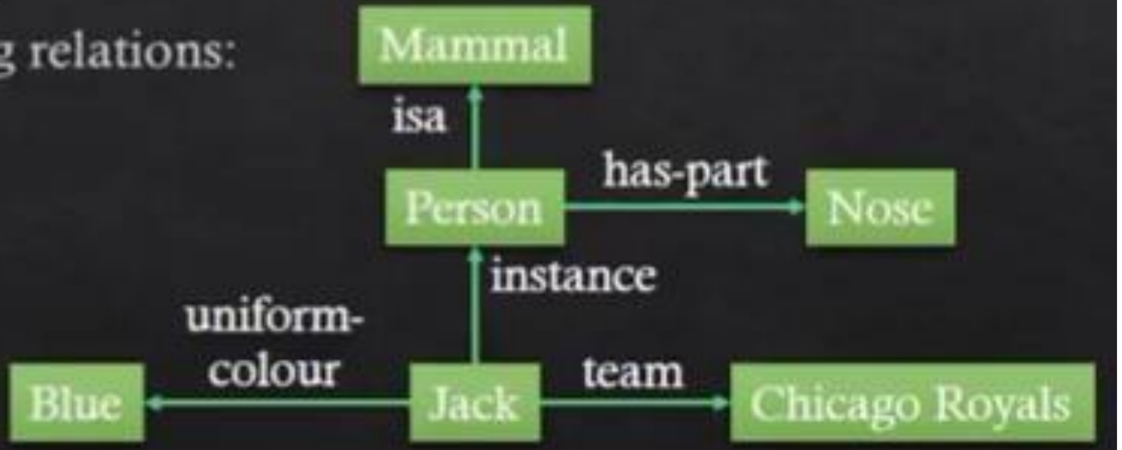
Semantic Nets

- Advantages of Slot-and-Filler structures
  - They support inheritance,
  - They support monotonic as well as non-monotonic reasoning effectively.
  - They make it easy to describe properties of relations.
  - Since they follow Object-Oriented approach, there is modularity

- Slot and filler structures are of two types:
  - Weak Slot-and-Filler Structures: Very little importance is given to the specific knowledge the structure should contain
  - Strong Slot-and-Filler structures: Specific commitments are made to the context of the representation.

Semantic Nets

- Semantics: Meaning of words
- Net: Network.
- In Semantic Nets, the meaning of a concept comes from the ways in which it is connected to other concepts.
- Information is represented as a set of nodes connected to each other by a set of labelled arcs, which represent relationships among nodes.
- In our example, we have represented the following relations:
  1. isa(Person, Mammal)
  2. has-part(Person, Nose)
  3. instance(Jack, Person)
  4. team(Jack, Chicago-Royals)
  5. uniform-color(Jack, Blue)

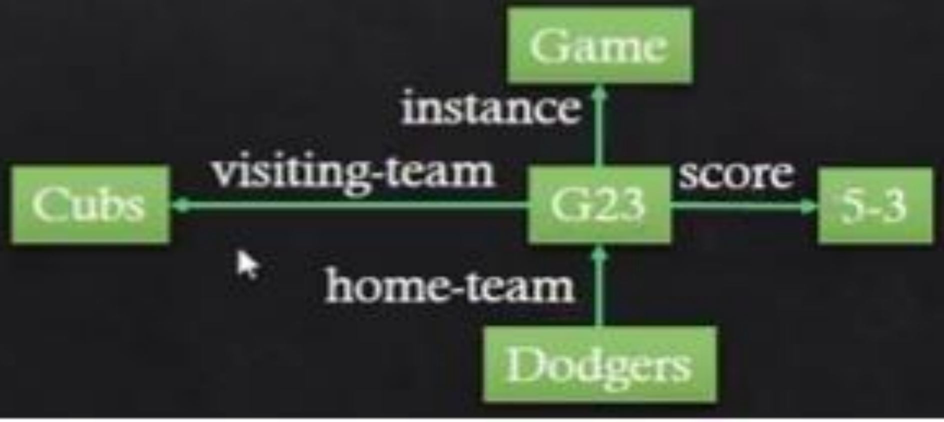


- But, from the same network, we could use inheritance to derive the additional relation:
  6. has-part(Jack, Nose)

Representing N-place Predicates using Semantic Nets

- Single Place Predicate:
  - Marcus is a man
    - man(Marcus)
    - instance(Marcus, Man)

- N-Place Predicate:
  - score(Cubs, Dodgers, 5-3)
    - three place predicate
      - visiting team, Cubs
      - home team, Dodgers
      - Score
    - Represent the game as a base class
    - G23 as instance of Game, with attributes:
      - Visiting team
      - Home team
      - Score
    - Game can be a class, instance can be G23



Representing Declarative Statement using Semantic Nets

- A declarative statement is a statement that states a fact.
  - Prabha likes riding a bike.
  - Vikram exchanges good books.
  - Pratima sang a classical song.
- Consider the statement: John gave the book to Mary
- Organise, the information into base class, derived class, their attributes, the values of those attributes, objects, their attributes and their values.

Representing Declarative Statement using Semantic Nets

- Consider the statement: John gave the book to Mary
- The event is an instance of the act of giving.
  - Give: class node
  - EV1: object of class give
  - John is the doer or the agent of the instance of giving.
  - The receiver of the instance of giving is Mary: beneficiary. The value of beneficiary attribute is Mary.
  - the item given is an instance of a class of books. That means we'll have to create a class node called book and its instance called say BK23 and this instance is the filler of the attribute item

Frame Based System

- When an agent encounters a new situation, it will need to retrieve information to act rationally in that situation. This information is likely to be multi-faceted and hierarchical. One way of structuring the knowledge is in terms of frames.
- These are frameworks consisting of slots.
- Each slot containing information in various representations like logical sentences, production rule, frame etc.
- Each framework represents a stereotypical object or situation.
- Whenever an agent encounters an object or situation which fits the stereotype, agent retrieve the framework and change some default information or fill the blank information.

Frame Representation

- Some of the information is procedural, so that when a blank is filled in with certain values, a procedure must be carried out.
- In this way frame dictate how to act rationally in situations.
- Types of Information store in slot

➢ Information for choosing the frame

➢ Information about relationships between frames.

➢ Procedures to be carried out.

➢ Default information.

➢ Blank slots.

Frame Representation

- Information for choosing the frame
- ➢ It may also be information about situations or descriptors for the stereotype the frame represents.
- ➢ E.g. name ,id
- Information about relationships between frames.
- ➢ Two frames should never be considered at the same time whether this frame is a generalisation or specialisation of another frame.
- Procedures to be carried out.
- ➢ It is rational action an agent should do in a situation where a particular value for a slot has been identified.
- Default information.
- ➢ It's the values when certain information required for the frame is missing.
- ➢ Default information is used in choosing actions until more specific information is found.
- Blank slots.
- ➢ These are flagged to be left blank unless required for a particular task.

FRAMES

- **FRAMES** :- means of representing common sense knowledge. Knowledge is organized into small packets called "Frames". All frames of a given situation constitute the system.
- A frame can be defined as a structure that has slots for various objects & a collection of frames consist of expectation for a given situation.
- Frame are used to represent two types of knowledge viz. declarative/factual and procedural, declarative & procedural Frames: -
- A frame that merely contains description about objects is call a declarative type/factual situational frame.

Name : Computer Centre	
A/c	Stationary cupboard
Computer	Dumb terminals
Printer	

Name of the frame

Slots in the frame

- Frames which have procedural knowledge embedded in it are called action procedure frames. The action frame has the following slots.
- Actor slot which holds information @ who is performing the activity.
- Source Slot hold information from where the action has to begin.
- Destination slot holds information about the place where action has to end.
- Task slot This generates the necessary sub frames required to perform the operation.

<table>
<tr><td colspan="6">Name : Cleaning the ict of carburetor</td></tr>
<tr><td colspan="6">Actor</td></tr>
<tr><td colspan="6">Expert
Object
Source Destination</td></tr>
<tr><td colspan="3">Scooter</td><td colspan="3">Scooter</td></tr>
<tr><td colspan="6">Task 1 Task 2
Task 3</td></tr>
<tr><td colspan="2">Remove Carburetor</td><td colspan="2">Clean Nozzle</td><td colspan="2">Fix Carburetor</td></tr>
</table>