Unit-5

Knowledge Inference

Knowledge Inference

- The knowledge base stored facts about the world. The inference engine applied logical rules to the knowledge base and deduced new knowledge. This process would iterate as each new fact in the knowledge base could trigger additional rules in the inference engine.
- Inference engines work primarily in one of two modes either special rule or facts:

➢ 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 restricted and 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 ∧ q → k.

Knowledge Inference

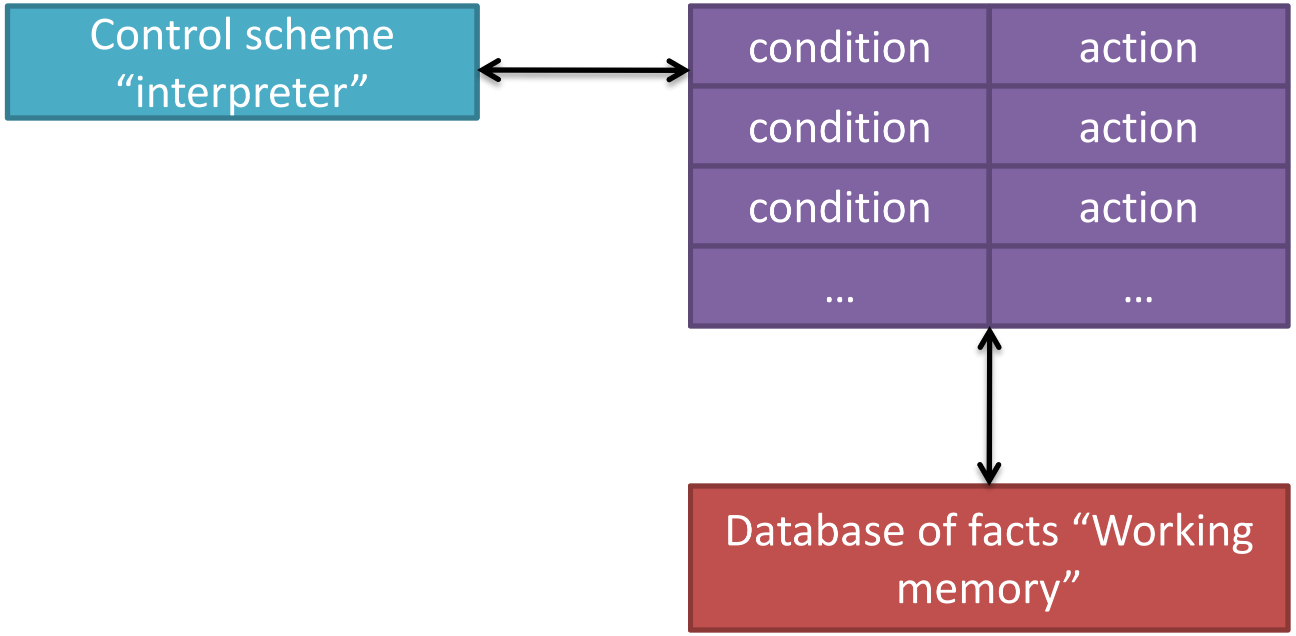
- In propositional logic, resolution by forward/backward chaining
  - ➢ Forward: Start from knowledge to reach query
  - ➢ Backward: Start from query and go back
- In FOL, substitute variables to get propositions
  - ➢ Use lifting and unification to resolve variables
- Logic programming: Prolog, LISP, Haskell

Prolog and LISP

- **Prolog**: Most widely used logic language.
  - ➢ Rules are written in backwards:

    criminal (X) :− american(X), weapon(Y), sells (X, Y, Z), hostile (Z)
  - ➢ Variables are uppercase and constants lowercase.
  - ➢ Because of complexity, often compiled into other languages like: Warren Abstract Machine, LISP or C. Language makes it easy to contruct lists, like LISP.
- **LISP**: LISt Processing language: primary data structure is lists.
  - ➢ Lisp is used for AI because can work with symbols Examples: computer algebra, theorem proving, planning systems, diagnosis, rewrite systems, knowledge representation and reasoning, logic languages, machine translation, expert systems, . . . It is a functional programming language, as opposed to a procedural or imperative language

Rule-Based Systems



Rule n:
IF condition1
condition2
…
antecedent

THEN
action1
action2
…
consequent

- 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.

Three Phases

1. Match phase
   - Match left side of rules (antecedents) with facts in working memory
   - Unification
2. Conflict resolution
   - Of all the rules that are triggered, decide which rule to fire.
   - Some strategies include:
     - Do not duplicate rule with same instantiated arguments twice
     - Prefer rules that refer to recently created WM elements
     - Prefer rules that are more specific (antecedents are more constraining)
     - Prefer Mammal(x) & Human(x) -> add Legs(x,2) over Mammal(x) -> add Legs(x,4)
   - If rules are ranked, fire according to ranking
3. Act phase
   - Add or delete facts to WM as specified by rule consequent

Forward Chaining

- When a new fact is added to the KB
  For each rule such that ``matches'' (unifies with) the premise
  If the other premises are **known**
  Then add the conclusion to the KB and continue chaining
- Forward chaining is **data-driven**
  e.g., inferring properties and categories from percepts

Forward Chaining

- Forward chaining 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.

Forward Chaining

- A simple example of forward chaining can be explained in the following sequence.
- A
- A->B
- ➢ B
- A is the starting point. A->B represents a fact. This fact is used to achieve a decision B.

Example: Tom is running (A)

- If a person is running, he will sweat (A->B)
- Therefore, Tom is sweating. (B)
- A DENDRAL expert system is a good example of how forward chaining is used in artificial intelligence. DENDRAL is used in the prediction of the molecular structure of substances.

Forward Chaining

- Deducing the chemical structure starts by finding the number of atoms in every molecule. The mass spectrum of the sample is then used to establish the arrangement of the atoms. We can summarize these steps as follows.
- The chemical formula is determined (the number of atoms in every molecule).
- The spectrum machine is used to form mass spectrums of the sample.
- The isomer and structure of the chemical are identified.
- In this example, the identification of the chemical structure is the endpoint. In the DENDRAL expert system, a generate and test technique is employed.
- There are two elements in the generator: a synthesiser and structural enumerator. The synthesiser plays the role of producing the mass spectrum. The structural enumerator identifies the structure of substances and prevents redundancy in the generator.

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, such as CLIPS, business, planning, monitoring, control and interpretation application, automated inference engines, theorem proofs and production rule systems.

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:
- 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)
- Country A has some missiles. **there exist 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)
- All of the missiles were sold to country A by Robert.
  forall p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)
- Missiles are weapons.
  Missile(p) → Weapons (p) ....(5)

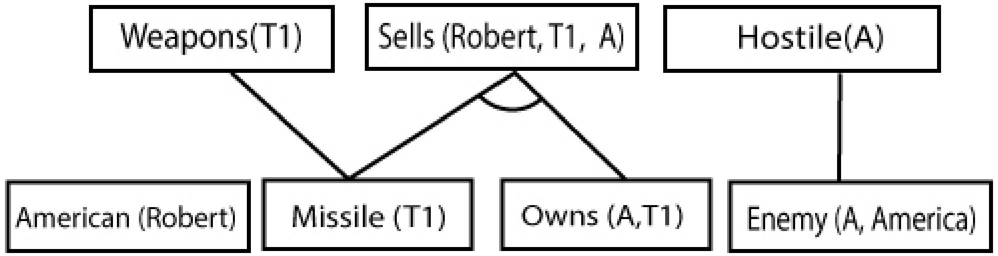
Forward Chaining

- Enemy of America is known as hostile.
  Enemy(p, America) →Hostile(p)   ....(6)
- Country A is an enemy of America.
  Enemy (A, America)   ....(7)
- Robert is American
  American(Robert).   ....(8)

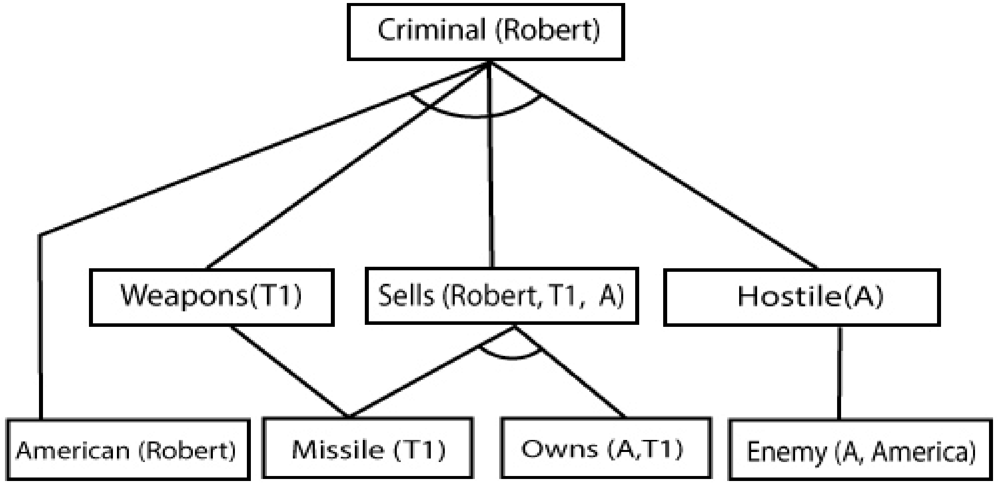
Forward Chaining

- **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).

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

Forward Chaining

- Reason FORWARD from facts/rules to (hopefully) a needed goal
- Use modus ponens to generate new facts
- Rule antecedents are compared with facts from database
- If match, add consequents to database
- Repeat as long as needed
- Forward chaining is “data driven”

Match Z1 & F1
Add: Stretch is a mammal

Match Z8/1 & F7 (?x Stretch)
Match Z8/2 & F2
Add: Stretch is an ungulate

Note: Z5/1, Z6/1, Z7/1 would be matched before Z8/1

Match Z11/1 & F8 (?x Stretch)
Match Z11/2 & F3
Match Z11/3 & F4
Match Z11/4 & F5
Match Z11/5 & F6
Add: Stretch is a giraffe

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
  - If match consequents, antecedents become new sub goals
- 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

- The MYCIN expert system is a real life example of how backward chaining works. This is a system that's used in the diagnosis of bacterial infections. It also recommends suitable treatments for this type of infections.
- The knowledge base of a MYCIN comprises many antecedent-consequent rules, that enable the system to recognize various causes of (bacterial) infections. This system is suitable for patients who have a **bacterial infection**, but don't know the specific infection. The system will gather information relating to symptoms and history of the patient. It will then analyze this information to establish the bacterial infection.
- A suitable sequence can be as follows:

➢ The patient has a bacterial infection.

➢ The patient is vomiting.

Backward Chaining

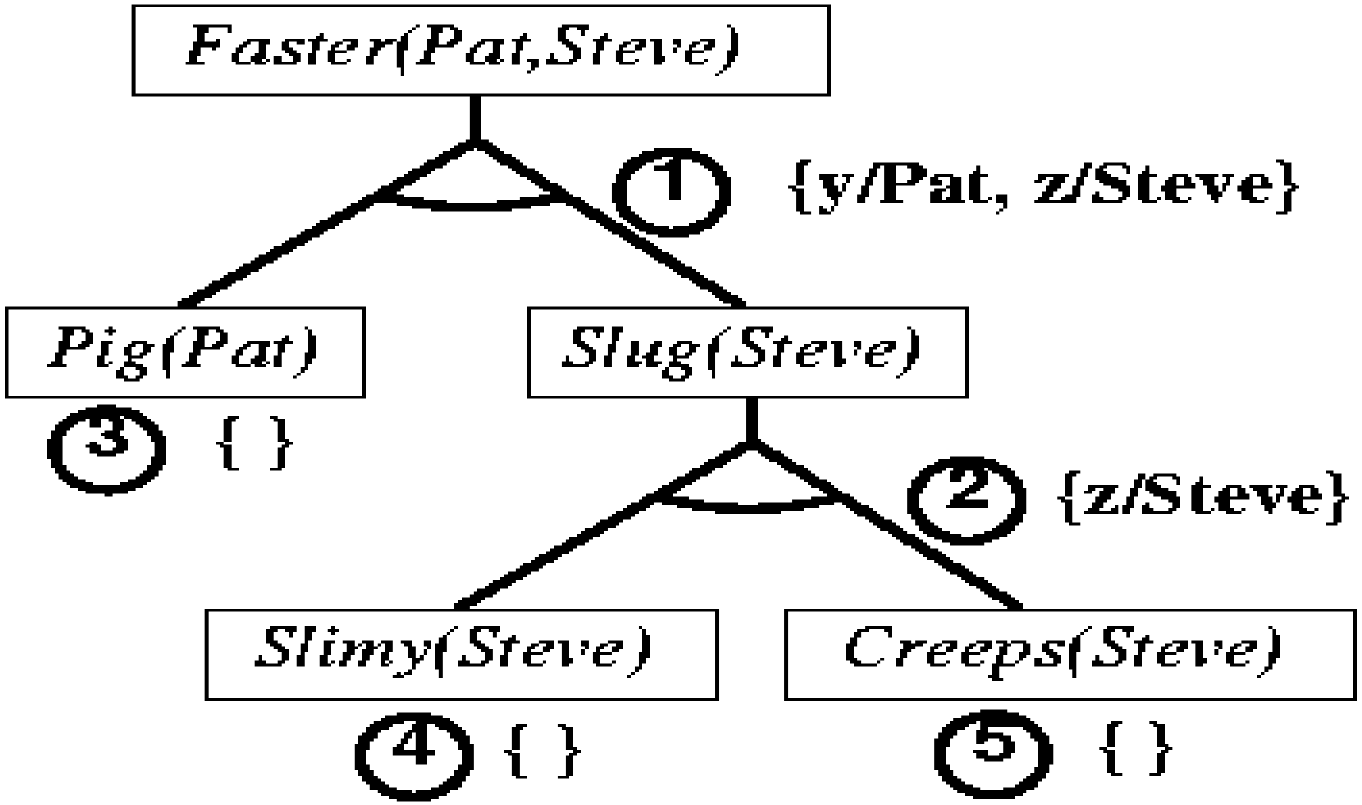
- He/she is also experiencing diarrhea and severe stomach upset.
- Therefore, the patient has **typhoid** (salmonella bacterial infection).
- The MYCIN expert system uses the information collected from the patient to recommend suitable treatment.
- The recommended treatment corresponds to the identified bacterial infection. In the case above, the system may recommend the use of ciprofloxacin.

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 Example

1. Pig(y) ^ Slug(z) -> Faster(y,z)
2. Slimy(z) ^ Creeps(z) -> Slug(z)
3. Pig(Pat)
4. Slimy(Steve)
5. Creeps(Steve)



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

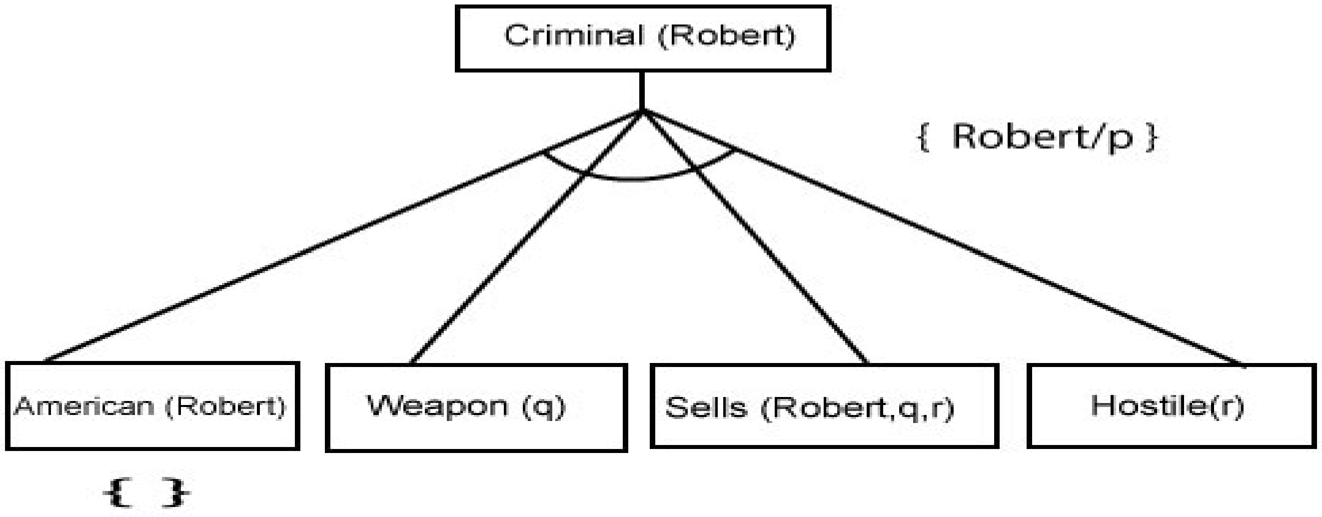
- Backward-Chaining proof:
- In Backward chaining, we will start with our goal predicate, which is **Criminal(Robert)**, and then infer further rules.
- **Step-1:**

➢ 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:**

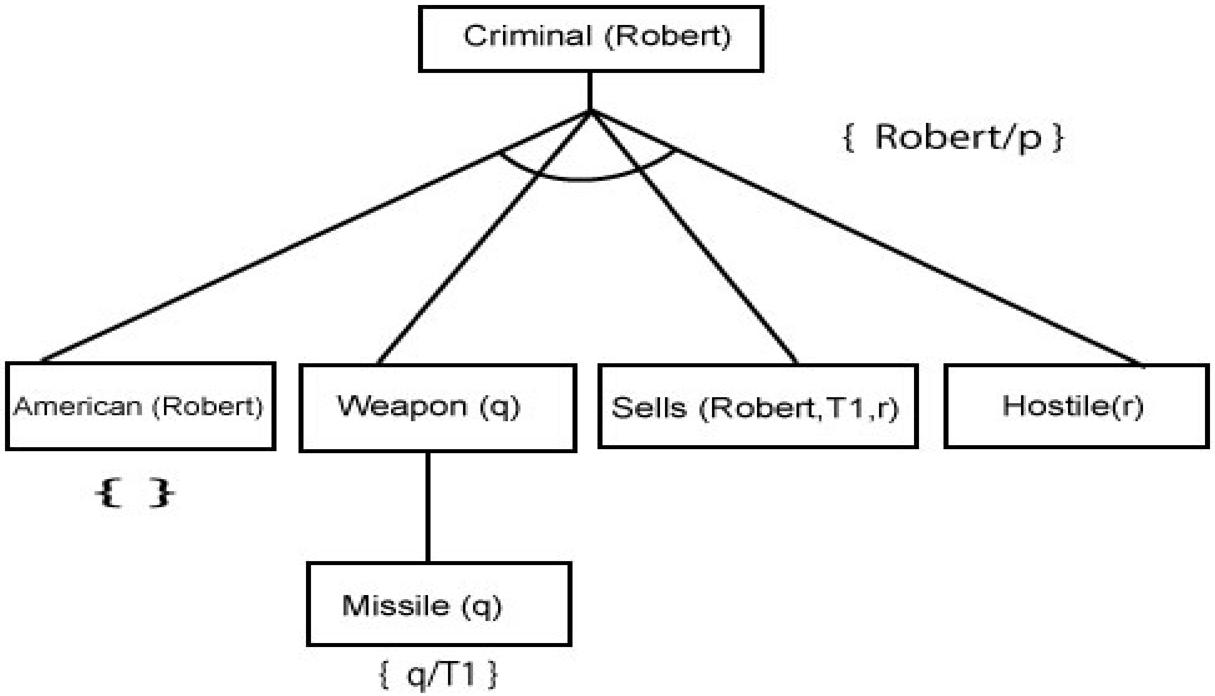
➢ At the second step, we will infer other facts form goal fact which satisfies the rules. So as we can see in Rule-1, the goal predicate Criminal (Robert) is present with substitution {Robert/P}. So we will add all the conjunctive facts below the first level and will replace p with Robert.

Backward Chaining



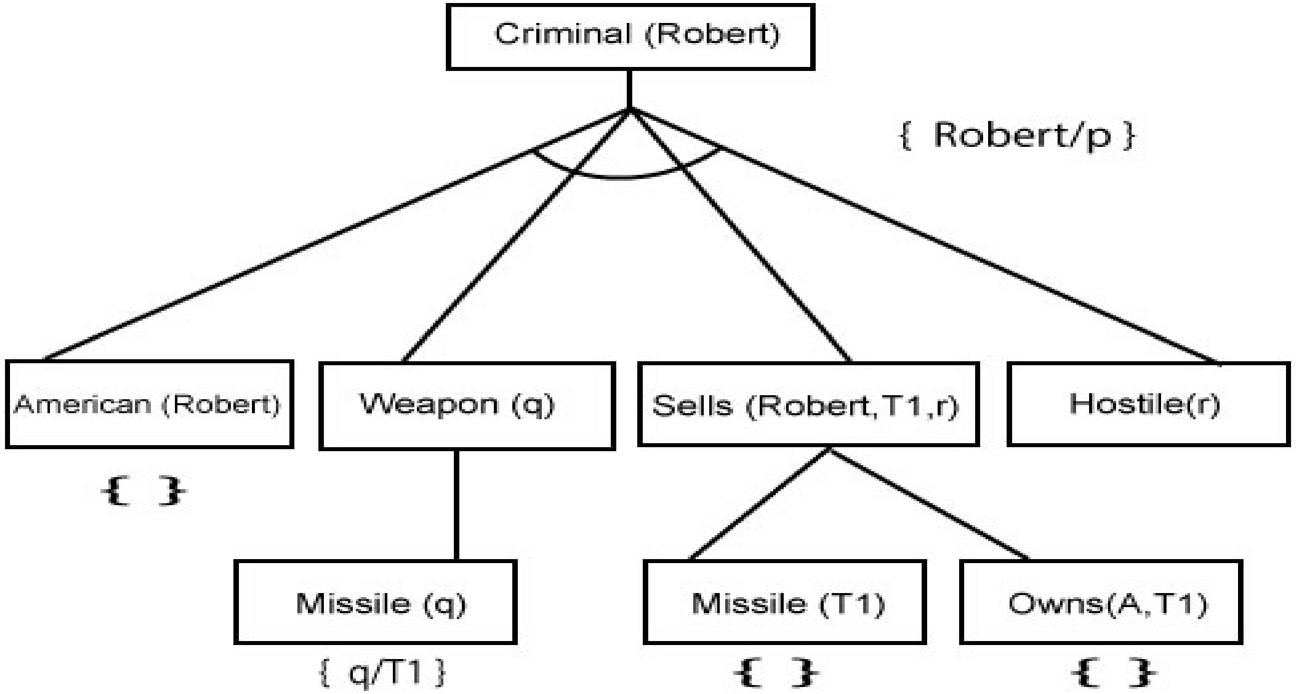
- **Step-3:**t At step-3, we will extract further fact Missile(q) which infer from Weapon(q), as it satisfies Rule-(5). Weapon (q) is also true with the substitution of a constant T1 at q.

Backward Chaining



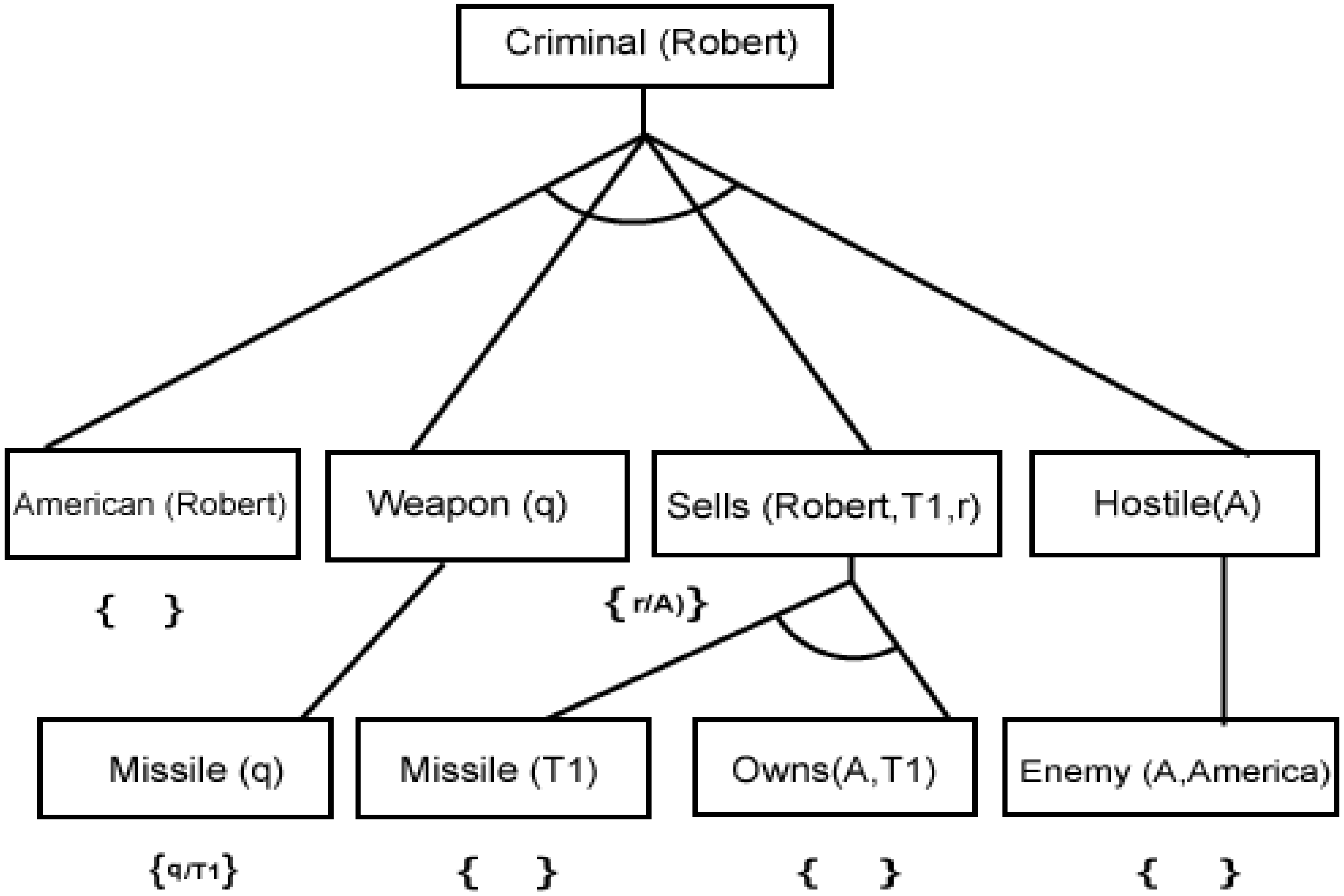
- **Step-4:**
- At step-4, we can infer facts Missile(T1) and Owns(A, T1) form Sells(Robert, T1, r) which satisfies the **Rule- 4**, with the substitution of A in place of r. So these two statements are proved here.

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



Backward Chaining Example

Goal 1: Stretch is a giraffe
- Match: Goal 1 and Z11/C (does not match with any facts)
- Subgoal 2: Stretch is an ungulate
- Subgoal 3: Stretch has long legs
- Subgoal 4: Stretch has long neck
- Subgoal 5: Stretch has tawny color
- Subgoal 6: Stretch has dark spots
    - Match: Subgoal 2 and Z8/C (does not match with any facts)
    - Subgoal 7: Stretch is a mammal
    - Subgoal 8: Stretch chews cud
        - Match: Subgoal 7 and Z1/C (does not match with any facts)
        - Subgoal 9: Stretch has hair
            - Match: Subgoal 9 and F1
                - Subgoals 9, 7, met
            - Match: Subgoal 8 and F2
                - Subgoals 8, 2 met
    - Match: Subgoal 3 and F3
        - Subgoal 3 met
    - Match: Subgoal 4 and F4
        - Subgoal 4 met
    - Match: Subgoal 5 and F5
        - Subgoal 5 met
    - Match: Subgoal 6 and F6
        - Subgoal 6 met, Goal 1 met

Forward Chaining vs. Backward Chaining

- High fan out - use backward chaining

1) Human(Albert)
2) Human(Alfred)
3) Human(Barry)
4) Human(Charlie)

   ...

50) Human(Highlander)

   ...

100) Human(Shaun)

   ...

500) Human(Zelda)
501) Human(x) -> Mortal(x)
502) Mortal(x) -> CanDie(x)

Can we kill Shaun Connery?

- FC

  503) Mortal(Albert)
  504) Moral(Alfred)
  …
  1003) CanDie(Albert)
  1004) CanDie(Alfred)
  …
  1100) CanDie(Shaun)

- BC
  - Prove: CanDie(Shaun)
    - Match: Goal and 502/C
    - Prove: Mortal(Shaun)
      - Match: Mortal(Shaun) and 501/C
      - Prove: Human(Shaun)
        - » Match: Human(Shaun) and 100
    - Done!

Forward Chaining vs. Backward Chaining

- Forward chaining may generate a lot of useless facts

If ?x has feathers
Than ?x can fly

If ?x has feathers
Then ?x is a bird
…

1 condition – many actions

If ?x has feathers
Then ?x can be used as a pen

Use BC!

Forward Chaining vs. Backward Chaining

- High fan in – use forward chaining
- Potential problem with BC
  - If many sub goals, each must be examined

If ?x has feathers
?x is brown
?x sings @#!@#! Patterns
?x sleeps on one foot
?x makes good pot pie
?x lives in tree
Then ?x is bird

- Many conditions – 1 action
- If time is crucial and you only have 1 goal to prove, use BC
- If you have extra time and want to be prepared for future questions, use FC to generate all possible facts
- Or use bi-directional search

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. Or we can say, "**Reasoning is a way to infer facts from existing data**." It is a general process of thinking rationally, to find valid conclusions.

- Human reasoning capability are divided into three areas:
- Mathematical reasoning-axioms, definitions, theorems, proofs
- Logical reasoning – deductive, inductive, abductive
- Non-logical reasoning – linguistic, language
- These 3 areas of reasoning are in every human being but the ability level depends on education, environment and genetics.
- Approaches to reasoning- Symbolic, statistical and fuzzy logic reasoning

Reasoning in AI

- To help machines perform human-level functions, artificial intelligence and its various subfields like machine learning, deep learning, natural language processing, and more 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. Additionally, reasoning can be performed either in a formal and informal manner or top-down or bottom-up approaches, depending on the way machines handle uncertainties and partial truths. Like Probabilistic Reasoning in Artificial Intelligence allows machines to deal with and represent uncertain knowledge and information.
- Reasoning, therefore, is the most fundamental capabilities associated with general intelligence, whether human or artificial, which enables both humans and machines to generate knowledge not available prior to the act of generation.

Role of Reasoning in Knowledge-based Systems

- A prominent form of Artificial Intelligence, Knowledge-based systems use knowledge from human experts to perform decision making and problem-solving. This is made possible by the knowledge base and inference engine. Moreover, it uses various reasoning techniques to deal with uncertainties in data and information, present in the knowledge base.
- Among the various types of reasoning, deductive reasoning and non-monotonic reasoning are used by knowledge-based systems to solve problems. Moreover, it helps with the implementation of knowledge-based systems and artificial intelligence in machines, enabling them to perform tasks that require human-level intelligence and mental processes.
- In short, like humans, machines use reasoning, along with knowledge representation, logic, and learning for analysis, problem-solving, making conclusions, and more

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 $\neg$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

- The world is an uncertain place, often the knowledge is imperfect which causes uncertainty, Therefore reasoning must be able to operate under uncertainty.
- **Uncertain inputs**
- ➢ Missing data
- ➢ Noisy data
- **Uncertain knowledge**
- ➢ Multiple causes lead to multiple effects
- ➢ Incomplete enumeration of conditions or effects
- ➢ Incomplete knowledge of causality in the domain
- ➢ Probabilistic/stochastic effects
- **Uncertain outputs**
- ➢ Abduction and induction are inherently uncertain
- ➢ Default reasoning, even in deductive fashion, is uncertain
- ➢ Incomplete deductive inference may be uncertain

Reasoning under uncertainty

- The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs. Or we can say, "**Reasoning is a way to infer facts from existing data**." It is a general process of thinking rationally, to find valid conclusions.
- **Types of Reasoning**

➢ Non Monotonic Reasoning

➢ Monotonic Reasoning

➢ Common Sense Reasoning

➢ Deductive reasoning

➢ Inductive reasoning

➢ Abductive reasoning

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.

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.

Monotonic Reasoning

- Standard type of logic
  - If proven true, will be true forever
  - Facts provided can’t be modified
- Doesn’t always fit in real life.
  - Sidra is in Doha and Doha is in Qatar, so Sidra is in Qatar.
  - Sidra can always take a trip to United States

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 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

- New facts can be added
- Current facts can be modified
- Conclusion can change

➢ If A → B before new fact

➢ Conclusion might change after new fact

- **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

- **Real life usage** : Used in artificial intelligent systems
- ➢ For its adaptability
- ➢ Adding, removing and modifying facts
- ➢ To reach appropriate conclusions for appropriate scenarios
- **Application** : Consider an example that can't be handled by monotonic logic
- ➢ Birds can fly (Seems logical, right?)
- **Exceptions?** • Ostrich, Penguins
- Bird(x) → Flies(x)
- How to handle exceptions? • Bird(x) ^ ¬Abnormal(x) → Flies(x)
- ➢ Through nonmonotonic logic, we handle exceptions

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.

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) v 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) V 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) ∨ 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 Knowledge base using GCWA looks like this:
  - Single(Jim) ∨ 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.
- Deductive reasoning is a type of propositional logic in AI, and it requires various rules and facts. 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.

- Theory
- Hypothesis
- Patterns
- 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, Inductive reasoning uses historical data to produce a generic rule whose conclusion is supported by the premises. Also known as cause-effect reasoning, this second type of reasoning is exploratory in nature and allows for uncertain but likely results.
- Like Deductive reasoning, inductive reasoning also follows a set of steps to perform reasoning, these are:

➢ Observation

➢ Pattern

➢ Hypothesis

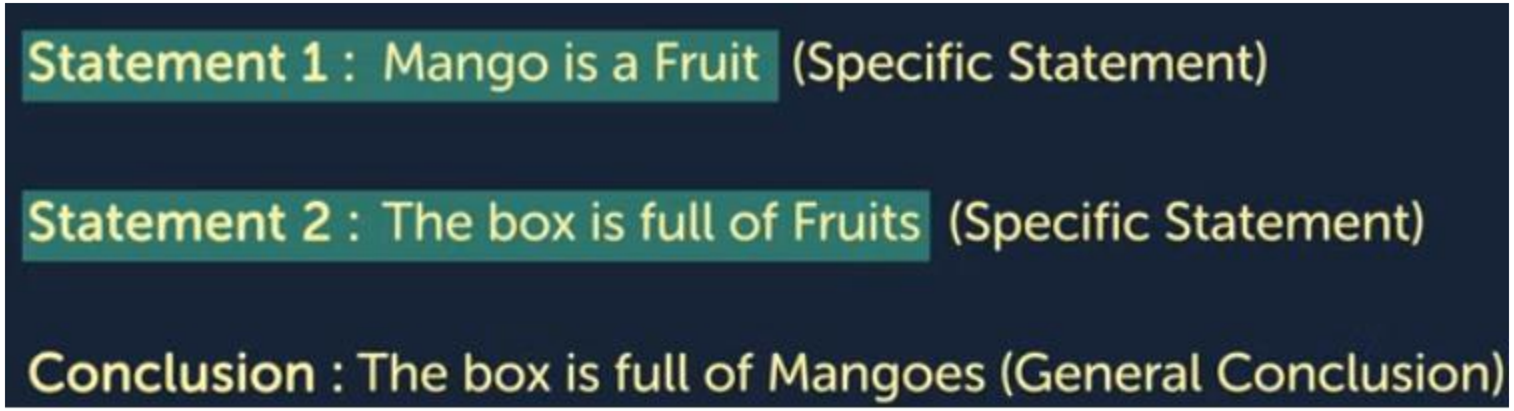
➢ Theory

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)



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)

- **Inductive reasoning** frequently used in mathematics by observing patterns that exist in a particular case; we deduce the general conclusion from that outcome.
- 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**
- **Abductive reasoning: taking your best shot**

Inductive vs deductive 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 five; four apples and one orange, therefore there are 16 apples and four oranges in the basket.

➢ **Predictive Inductive Reasoning Example**:

- Most baseball players become coaches. Eduardo is a baseball player, so he'll become a coach.

➢ **Statistical Syllogism Inductive Reasoning Example**:

- 95% of Oxford graduates went on to get PhD's; Rocky graduated from Oxford, therefore he's going to get a PhD.

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.

Inductive research approach

- When there is little to no existing literature on a topic, it is common to perform inductive research because there is no theory to test. The inductive approach consists of three stages:
- **Observation**
  - A low-cost airline flight is delayed
  - Dogs A and B have fleas
  - Elephants depend on water to exist
- **Observe a pattern**
  - Another 20 flights from low-cost airlines are delayed
  - All observed dogs have fleas
  - All observed animals depend on water to exist
- **Develop a theory**
  - Low cost airlines always have delays
  - All dogs have fleas
  - All biological life depends on water to exist

Inductive research approach

- **Limitations of an inductive approach**

➢ A conclusion drawn on the basis of an inductive method can never be proven, but it can be invalidated.

- **Example**

  You observe 1000 flights from low-cost airlines. All of them experience a delay, which is in line with your theory. However, you can never prove that flight 1001 will also be delayed. Still, the larger your dataset, the more reliable the conclusion.

Deductive research approach

- When conducting deductive research, you always start with a theory (the result of inductive research). Reasoning deductively means testing these theories. If there is no theory yet, you cannot conduct deductive research.
- The deductive research approach consists of four stages:
- **Start with an existing theory**
  - Low cost airlines always have delays
  - All dogs have fleas
  - All biological life depends on water to exist
- **Formulate a hypothesis based on existing theory**
  - If passengers fly with a low cost airline, then they will always experience delays
  - All pet dogs in my apartment building have fleas
  - All land mammals depend on water to exist

Deductive research approach

- **Collect data to [test the hypothesis]**
  - Collect flight data of low-cost airlines
  - Test all dogs in the building for fleas
  - Study all land mammal species to see if they depend on water
- **Analyse the results: does the data reject or support the hypothesis?**
  - 5 out of 100 flights of low-cost airlines are not delayed = reject hypothesis
  - 10 out of 20 dogs didn’t have fleas = reject hypothesis
  - All land mammal species depend on water = support hypothesis

Deductive research approach

- **Limitations of a deductive approach**
  - ➢ The conclusions of deductive reasoning can only be true if all the premises set in the inductive study are true and the terms are clear.
- **Example**
  - ➢ All dogs have fleas (premise)
  - ➢ Benno is a dog (premise)
  - ➢ Benno has fleas (conclusion)
- Based on the premises we have, the conclusion must be true. However, if the first premise turns out to be false, the conclusion that Benno has fleas cannot be relied upon.

Abductive Reasoning

- An extension of deductive reasoning, abductive reasoning, is a form of logical inference that seeks theories to define observations using the simplest and likely explanation. Compared to inductive reasoning, abductive reasoning, which is also known as abductive inference or retroduction, is less rigorous and allows for most accurate suggestions and guesses. However, unlike deductive reasoning, it does not rectify the plausible conclusion and is mostly used in data certainties. It is associated with troubleshooting and decision making.

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:
- Incomplete observations -> Best Prediction (may be true)
  - 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.

Thank you.