Forward Chaining and backward chaining in AI

In artificial intelligence, forward and backward chaining is one of the important topics, but before understanding forward and backward chaining lets first understand that from where these two terms came.

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 $p \wedge q \rightarrow k$.

A. 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.

Properties of Forward-Chaining:

- o   It is a down-up approach, as it moves from bottom to top.

- o   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.

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

- o Forward -chaining approach is commonly used in the expert system, such as CLIPS, business, and production rule systems.

Consider the following famous example which we will use in both approaches:

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.

Facts Conversion into FOL:

- o 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)

- o 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)

- o All of the missiles were sold to country A by Robert.
  ?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)

- o Missiles are weapons.
  Missile(p) → Weapons (p) ....(5)

- o Enemy of America is known as hostile.
  Enemy(p, America) → Hostile(p) ....(6)

- o Country A is an enemy of America.
  Enemy (A, America) ....(7)

- o Robert is American
  American(Robert). ....(8)

Forward chaining proof:

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)**. All these facts will be represented as below.

| American (Robert) | Missile (T1) | Owns (A,T1) | Enemy (A, America) |
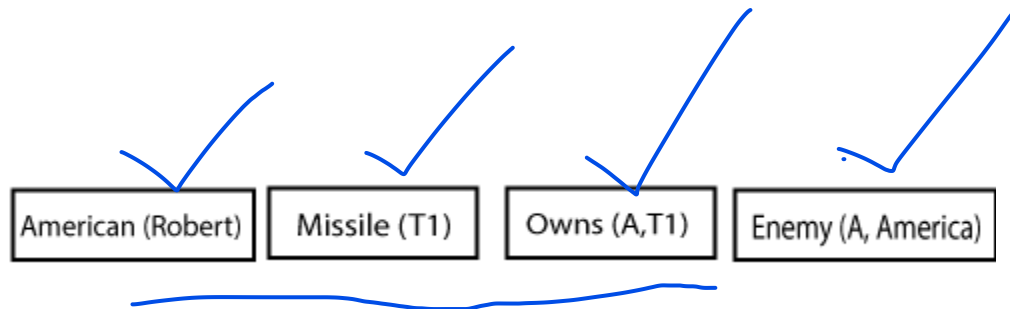
Step-2:

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.
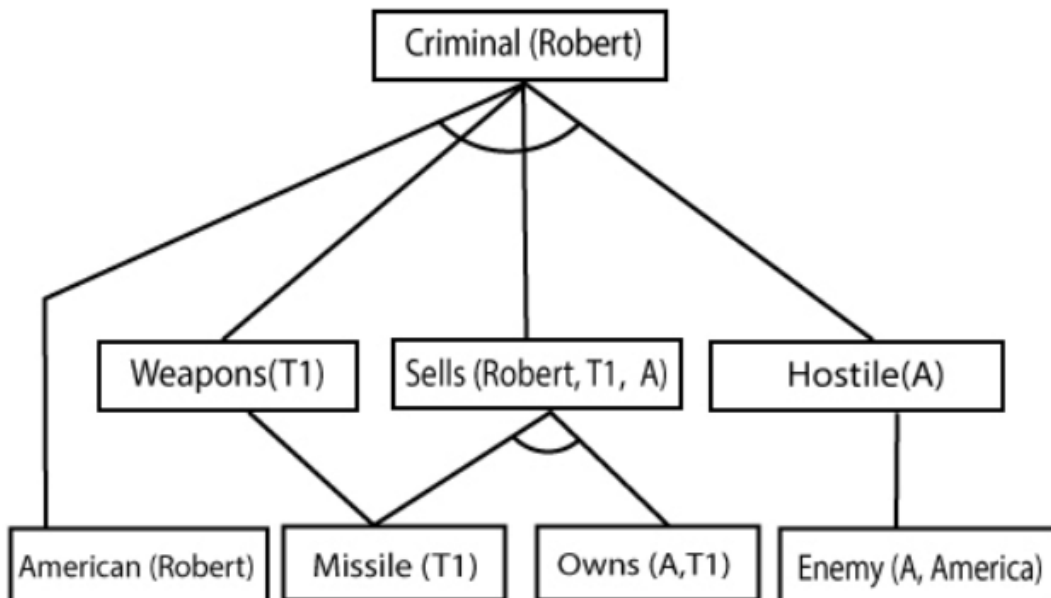
Rule-(2) and (3) are already added.

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) |
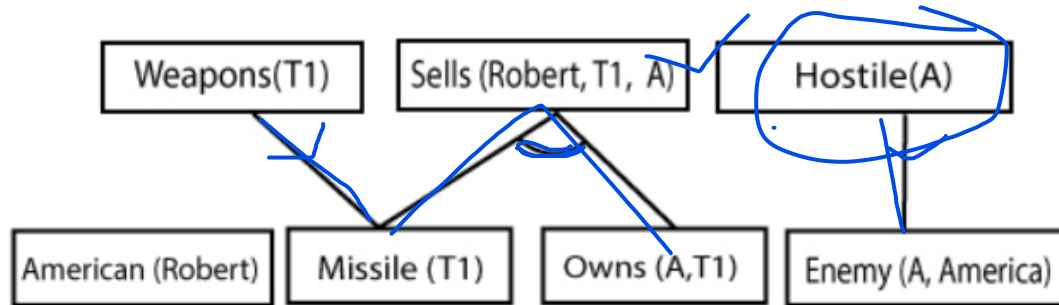
Step-3:

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.

Criminal (Robert)

| Weapons(T1) | Sells (Robert, T1, A) | Hostile(A) |

| American (Robert) | Missile (T1) | Owns (A,T1) | Enemy (A, America) |

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

B. 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.

Properties of backward chaining:

- o   It is known as a top-down approach.
- o   Backward-chaining is based on modus ponens inference rule.
- o   In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.
- o   It is called a goal-driven approach, as a list of goals decides which rules are selected and used.
- o   Backward -chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.
- o   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.

- o   **American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p) ...(1)**
  Owns(A, T1)   ....(2)
- o   **Missile(T1)**
- o   **?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A)   ......(4)**
- o   **Missile(p) → Weapons (p)   ....(5)**
- o   **Enemy(p, America) →Hostile(p)   ....(6)**
- o   **Enemy (A, America)   ....(7)**
- o   **American(Robert).   ....(8)**

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.

Criminal (Robert)

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.

Here we can see American (Robert) is a fact, so it is proved here.

Criminal (Robert)

{ Robert/p }

American (Robert)   Weapon (q)   Sells (Robert,q,r)   Hostile(r)

{ }
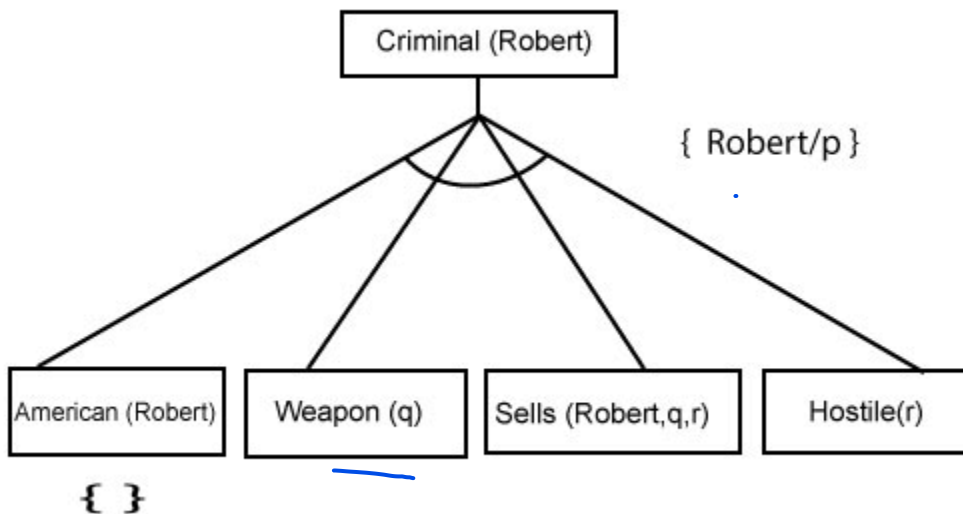
Step-3:t At step-3, we will extract further fact Missile(q) which infer from Weapon(q), as it satisfies Rule-(5). Weapon (q) is also true with the substitution of a constant T1 at q.
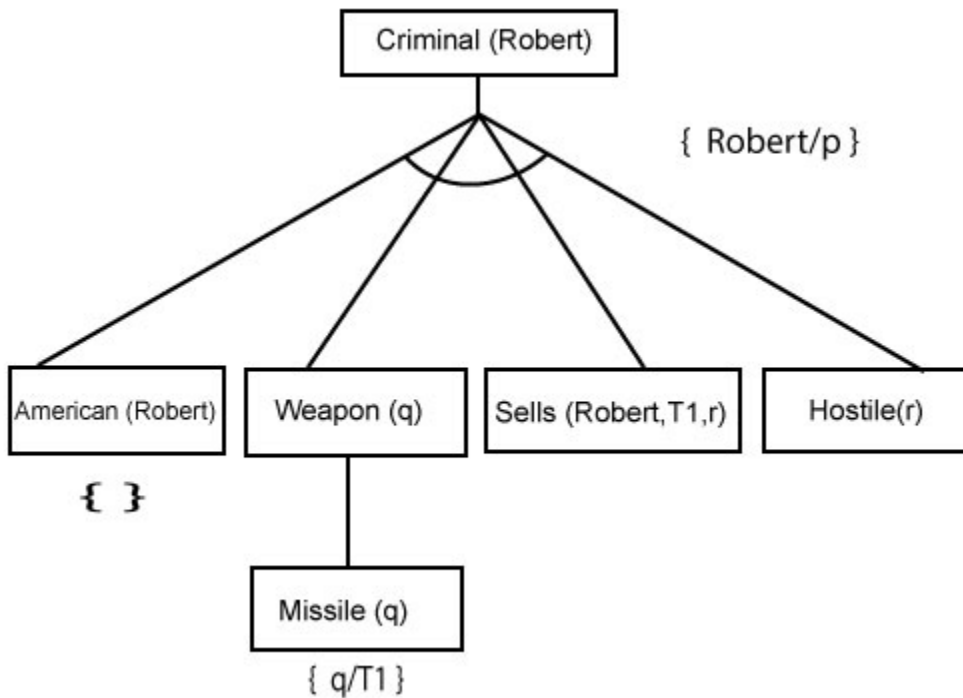
Step-4:

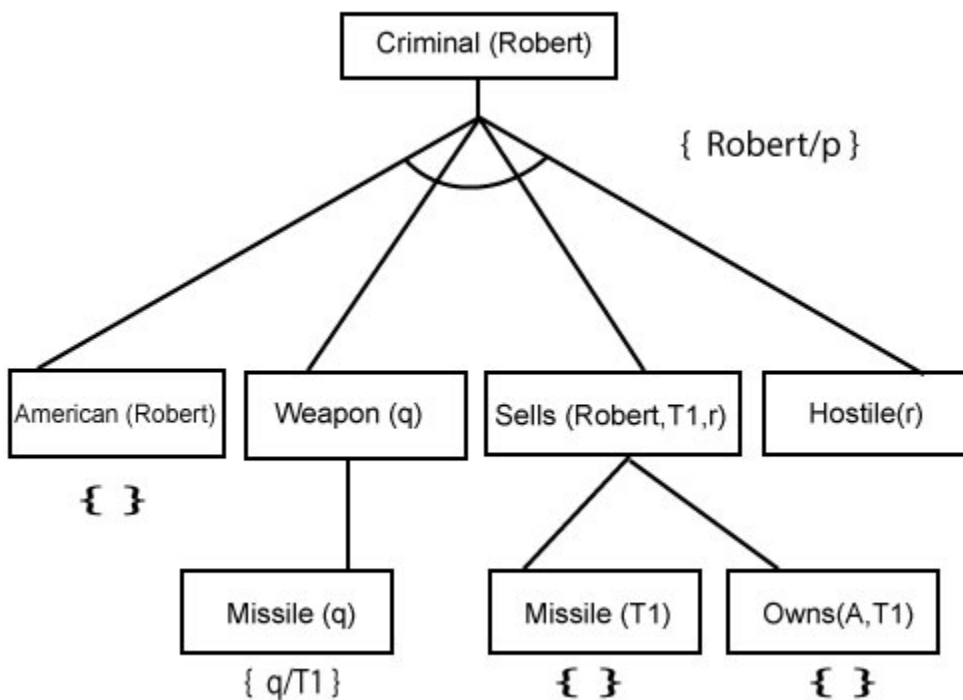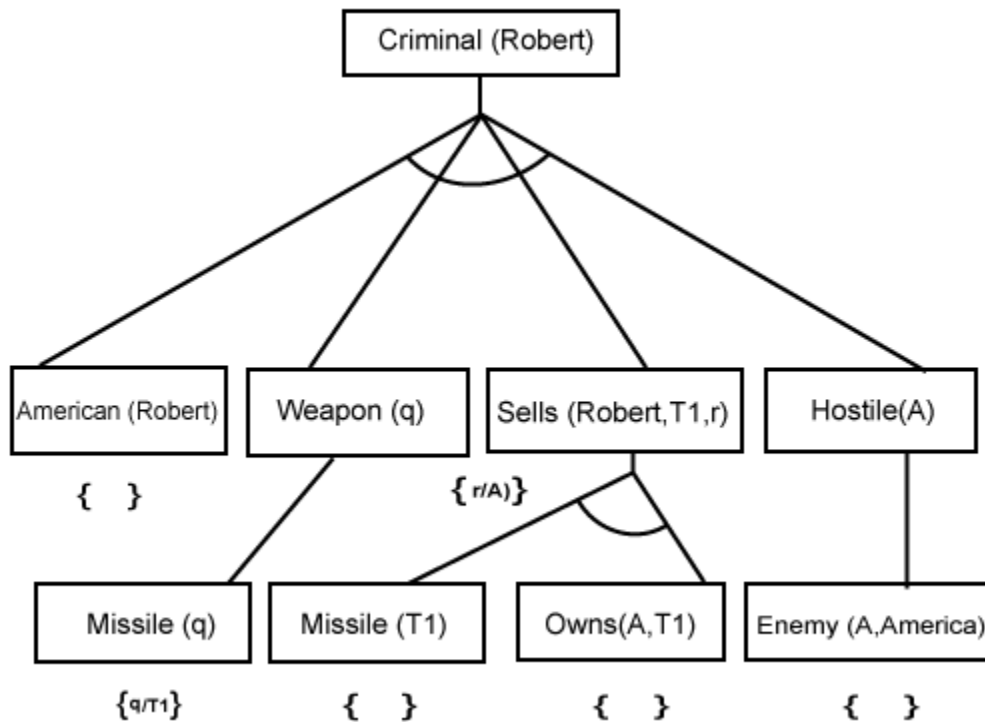At step-4, we can infer facts Missile(T1) and Owns(A, T1) form Sells(Robert, T1, r) which satisfies the **Rule- 4**, with the substitution of A in place of r. So these two statements are proved here.

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.

```
                        ┌─────────────────┐
                        │ Criminal (Robert)│
                        └─────────────────┘

┌────────────────┐  ┌──────────────┐  ┌──────────────────┐  ┌──────────────┐
│American (Robert)│  │  Weapon (q)  │  │Sells (Robert,T1,r)│  │  Hostile(A)  │
└────────────────┘  └──────────────┘  └──────────────────┘  └──────────────┘
      { }                                    { r/A)}

        ┌─────────────┐  ┌─────────────┐  ┌──────────────┐  ┌──────────────────┐
        │ Missile (q) │  │ Missile (T1)│  │  Owns(A,T1)  │  │Enemy (A,America) │
        └─────────────┘  └─────────────┘  └──────────────┘  └──────────────────┘
          {q/T1}             { }              { }                  { }
```

Difference between backward chaining and forward chaining

Following is the difference between the forward chaining and backward chaining:

- o Forward chaining as the name suggests, start from the known facts and move forward by applying inference rules to extract more data, and it continues until it reaches to the goal, whereas backward chaining starts from the goal, move backward by using inference rules to determine the facts that satisfy the goal.

- o Forward chaining is called a **data-driven** inference technique, whereas backward chaining is called a **goal-driven** inference technique.

- o Forward chaining is known as the **down-up** approach, whereas backward chaining is known as a **top-down** approach.

- o Forward chaining uses **breadth-first search** strategy, whereas backward chaining uses **depth-first search** strategy.

- o Forward and backward chaining both applies **Modus ponens** inference rule.

- o Forward chaining can be used for tasks such as **planning, design process monitoring, diagnosis, and classification**, whereas backward chaining can be used for **classification and diagnosis tasks**.

- Forward chaining can be like an exhaustive search, whereas backward chaining tries to avoid the unnecessary path of reasoning.

- In forward-chaining there can be various ASK questions from the knowledge base, whereas in backward chaining there can be fewer ASK questions.

- Forward chaining is slow as it checks for all the rules, whereas backward chaining is fast as it checks few required rules only.

S. No.	Forward Chaining	Backward Chaining
1.	Forward chaining starts from known facts and applies inference rule to extract more data unit it reaches to the goal.	Backward chaining starts from the goal and works backward through inference rules to find the required facts that support the goal.
2.	It is a bottom-up approach	It is a top-down approach
3.	Forward chaining is known as data-driven inference technique as we reach to the goal using the available data.	Backward chaining is known as goal-driven technique as we start from the goal and divide into sub-goal to extract the facts.
4.	Forward chaining reasoning applies a breadth-first search strategy.	Backward chaining reasoning applies a depth-first search strategy.
5.	Forward chaining tests for all the available rules	Backward chaining only tests for few required rules.
6.	Forward chaining is suitable for the planning, monitoring, control, and interpretation application.	Backward chaining is suitable for diagnostic, prescription, and debugging application.
7.	Forward chaining can generate an infinite number of possible conclusions.	Backward chaining generates a finite number of possible conclusions.
8.	It operates in the forward direction.	It operates in the backward direction.
9.	Forward chaining is aimed for any conclusion.	Backward chaining is only aimed for the required data.