INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND ITS APPLICATIONS

Code:19EAI232

Dr. Praveen Gupta
Asst. Professor
Department of Computer Science and Engineering,
GITAM Institute of Technology,
GITAM (Deemed to be University), Visakhapatnam(AP), India
Email Id: pgupta@gitam.edu , M: 97837 50898

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

LEARNING OUTCOMES:

After completion of this module, the student will be able to:

- define knowledge-based agents (L1)
- how to represent real world facts in propositional and first-order logic (L1)
- explain the wumpus world problem-solving process (L2)
- infer proofs in propositional and first-order logic (L2)
- contrast forward chaining and backward chaining (L2)

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

FIRST-ORDER LOGIC (FOL)

- First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- First-order logic is also known as Predicate logic or First-order predicate logic.
- First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.

FIRST-ORDER LOGIC (FOL)

- First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:
  - ✓ Objects: A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
  - ✓ Relations: It can be unary relation such as: red, round, is adjacent, or n-any relation such as: the sister of, brother of, has color, comes between
  - ✓ Function: Father of, best friend, third inning of, end of, ...
- As a natural language, first-order logic also has two main parts:
  - ✓ Syntax
  - ✓ Semantics

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

SYNTAX OF FIRST-ORDER LOGIC

- The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols.
- We write statements in short-hand notation in FOL.

BASIC ELEMENTS OF FIRST-ORDER LOGIC:

Constant	1, 2, A, John, Mumbai, cat,....
Variables	x, y, z, a, b,....
Predicates	Brother, Father, >,....
Function	sqrt, LeftLegOf, ...
Connectives	$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
Equality	==
Quantifier	$\forall$, $\exists$

SYNTAX OF FIRST-ORDER LOGIC

- ATOMIC SENTENCES

✓ Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.

✓ We can represent atomic sentences as Predicate (term1, term2, ..., term n).

Example:

Ravi and Ajay are brothers: => Brothers(Ravi, Ajay).

Chinky is a cat: => cat (Chinky).

SYNTAX OF FIRST-ORDER LOGIC

GITAM
(DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

- COMPLEX SENTENCES

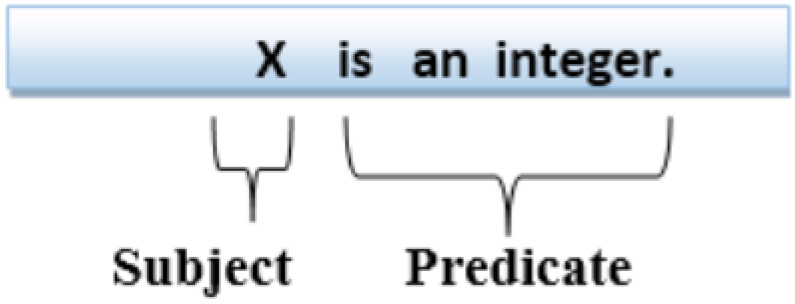
✓ Complex sentences are made by combining atomic sentences using connectives.

✓ First-order logic statements can be divided into two parts:

➢ Subject: Subject is the main part of the statement.

➢ Predicate: A predicate can be defined as a relation, which binds two atoms together in a statement.

✓ Consider the statement: "x is an integer.", it consists of two parts, the first part x is the subject of the statement and second part "is an integer," is known as a predicate.



QUANTIFIERS IN FIRST-ORDER LOGIC

- A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:
- Universal Quantifier, (for all, everyone, everything)
- Existential quantifier, (for some, at least one).

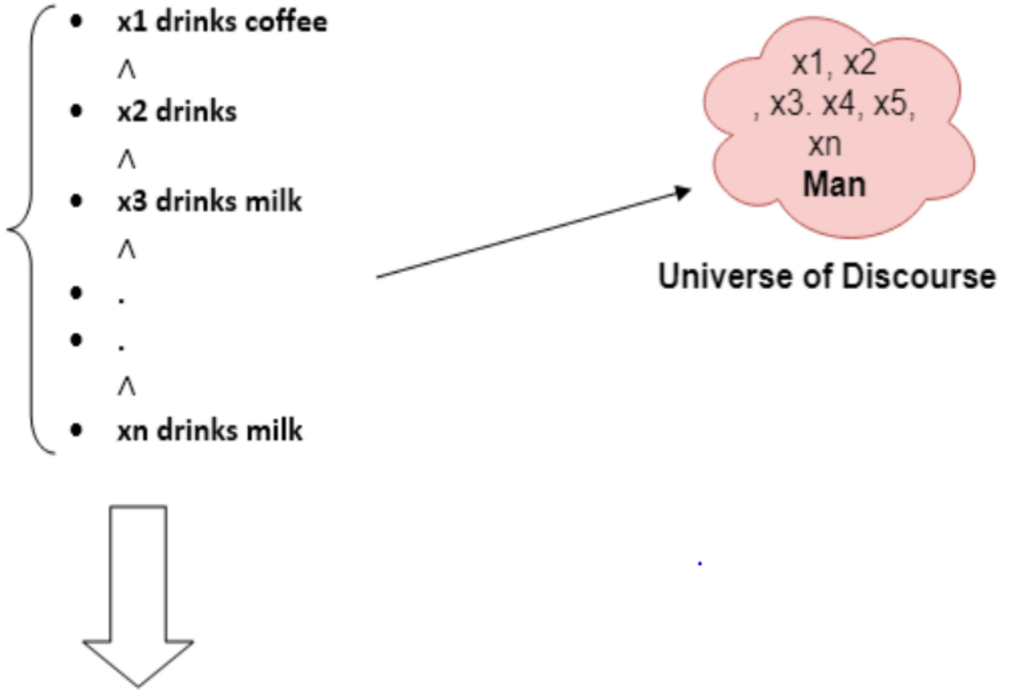
UNIVERSAL QUANTIFIER

GITAM (DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

- Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.
- The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.
- Note: In universal quantifier we use implication "→".
- If x is a variable, then ∀x is read as:
  - ✓ For all x
  - ✓ For each x
  - ✓ For every x.

All man drink coffee.

Let a variable x which refers to a cat so all x can be represented in UOD as below:



So in shorthand notation, we can write it as :

∀x man(x) → drink (x, coffee).

It will be read as: There are all x where x is a man who drink coffee.

EXISTENTIAL QUANTIFIER

- Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.
- It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.
- Note: In Existential quantifier we always use AND or Conjunction symbol (∧).
- If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:
  - There exists a 'x.'
  - For some 'x.'
  - For at least one 'x.'

Example:

Some boys are intelligent.

- **x1 is intelligent**
  V
- **x2 is intelligentV**
- **x3 is intelligent**
  V
- .
- .
  V
- **xn is intelligent**

Universe of Discourse

So in short-hand notation, we can write it as:

∃x: boys(x) ∧ intelligent(x)

It will be read as: There are some x where x is a boy who is intelligent.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

INFERENCE IN FIRST-ORDER LOGIC

- Inference in First-Order Logic is used to deduce new facts or sentences from existing sentences.
- Before understanding the FOL inference rule, let's understand some basic terminologies used in FOL.

1. Substitution:

- Substitution is a fundamental operation performed on terms and formulas.
- It occurs in all inference systems in first-order logic.
- The substitution is complex in the presence of quantifiers in FOL.
- If we write F[a/x], so it refers to substitute a constant "a" in place of variable "x".

Contd..

2. Equality:

✓ First-Order logic does not only use predicate and terms for making atomic sentences but also uses another way, which is equality in FOL.

✓ For this, we can use equality symbols which specify that the two terms refer to the same object.

Example: Brother (John) = Smith.

✓ As in the above example, the object referred by the Brother (John) is similar to the object referred by Smith.

✓ The equality symbol can also be used with negation to represent that two terms are not the same objects.

Example: $\neg(x=y)$ which is equivalent to $x \neq y$.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

DIFFERENTIATION BETWEEN PROPOSITIONAL LOGIC AND FIRST-ORDER LOGIC

- Propositional logic is an analytical statement which is either true or false.
- It is basically a technique that represents the knowledge in logical & mathematical form.
- There are two types of propositional logic;

  i. Atomic and

  ii. Compound Propositions.

Features of Propositional Logic

- Concerned with the subset of declarative sentences that can be classified as true or false.
- We call these sentences "statements" or "propositions".
- Paradoxes – statements that cannot be classified as true or false.
- Open sentences – statements that cannot be answered absolutely.

FACTS ABOUT PROPOSITIONAL LOGIC

- Since propositional logic works on 0 and 1 thus it is also known as ‘Boolean Logic’.
- Proposition logic can be either true or false it can never be both.
- In this type of logic, symbolic variables are used in order to represent the logic and any logic can be used for representing the variable.
- t is comprised of objects, relations, functions, and logical connectives.
- Proposition formula which is always false is called ‘Contradiction’ whereas a proposition formula which is always true is called ‘Tautology’.

FACTS ABOUT FIRST ORDER LOGIC

- FOL is known as the powerful language which is used to develop information related to objects in a very easy way.
- Unlike PL, FOL assumes some of the facts that are related to objects, relations, and functions.
- FOL has two main key features or you can say parts that are; ‘Syntax’ & ‘Semantics’.

KEY DIFFERENCES BETWEEN PL AND FOL

- Propositional Logic converts a complete sentence into a symbol and makes it logical whereas in First-Order Logic relation of a particular sentence will be made that involves relations, constants, functions, and constants.
- The limitation of PL is that it does not represent any individual entities whereas FOL can easily represent the individual establishment that means if you are writing a single sentence then it can be easily represented in FOL.
- PL does not signify or express the generalization, specialization or pattern for example ‘QUANTIFIERS’ cannot be used in PL but in FOL users can easily use quantifiers as it does express the generalization, specialization, and pattern.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

UNIFICATION

- Unification is a process of making two different logical atomic expressions identical by finding a substitution. Unification depends on the substitution process.
- It takes two literals as input and makes them identical using substitution.
- Let Ψ_1 and Ψ_2 be two atomic sentences and σ be a unifier such that, $\Psi_1\sigma = \Psi_2\sigma$, then it can be expressed as UNIFY(Ψ_1, Ψ_2).
- Example: Find the MGU for Unify{King(x), King(John)}

CONDITIONS FOR UNIFICATION

- Following are some basic conditions for unification:
- Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.
- Number of Arguments in both expressions must be identical.
- Unification will fail if there are two similar variables present in the same expression.

UNIFICATION ALGORITHM

GITAM (DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

ALGORITHM: UNIFY(Ψ1, Ψ2)

- Step. 1: If Ψ1 or Ψ2 is a variable or constant, then:
- a) If Ψ1 or Ψ2 are identical, then return NIL.
- b) Else if Ψ1is a variable,
- a. then if Ψ1 occurs in Ψ2, then return FAILURE
- b. Else return { (Ψ2/ Ψ1)}.
- c) Else if Ψ2 is a variable,
- a. If Ψ2 occurs in Ψ1 then return FAILURE,
- b. Else return {(Ψ1/ Ψ2)}.
- d) Else return FAILURE.
- Step.2: If the initial Predicate symbol in Ψ1 and Ψ2 are not same, then return FAILURE.

IMPLEMENTATION OF UNIFICATION ALGORITHM

Step.1: Initialize the substitution set to be empty.

Step.2: Recursively unify atomic sentences:

- Check for Identical expression match.
- If one expression is a variable vi, and the other is a term ti which does not contain variable vi, then:

  a. Substitute ti / vi in the existing substitutions

  b. Add ti /vi to the substitution setlist.

  c. If both the expressions are functions, then function name must be similar, and the number of arguments must be the same in both the expression.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.

- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

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

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

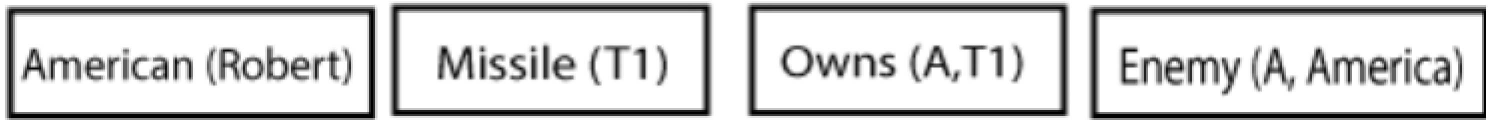
FORWARD CHAINING: EXAMPLE

- 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. ?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.
- ?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)
- Missiles are weapons. Missile(p) → Weapons (p) ....(5)
- 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 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.



FORWARD CHAINING PROOF

- Step-2:

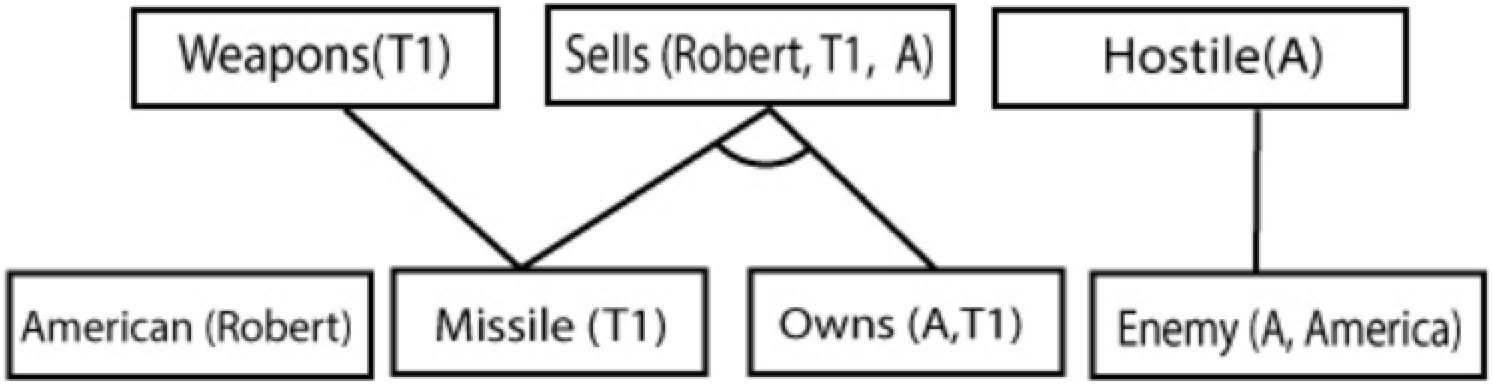
✓At the second step, we will see those facts which infer from available facts and with satisfied premises.

✓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).

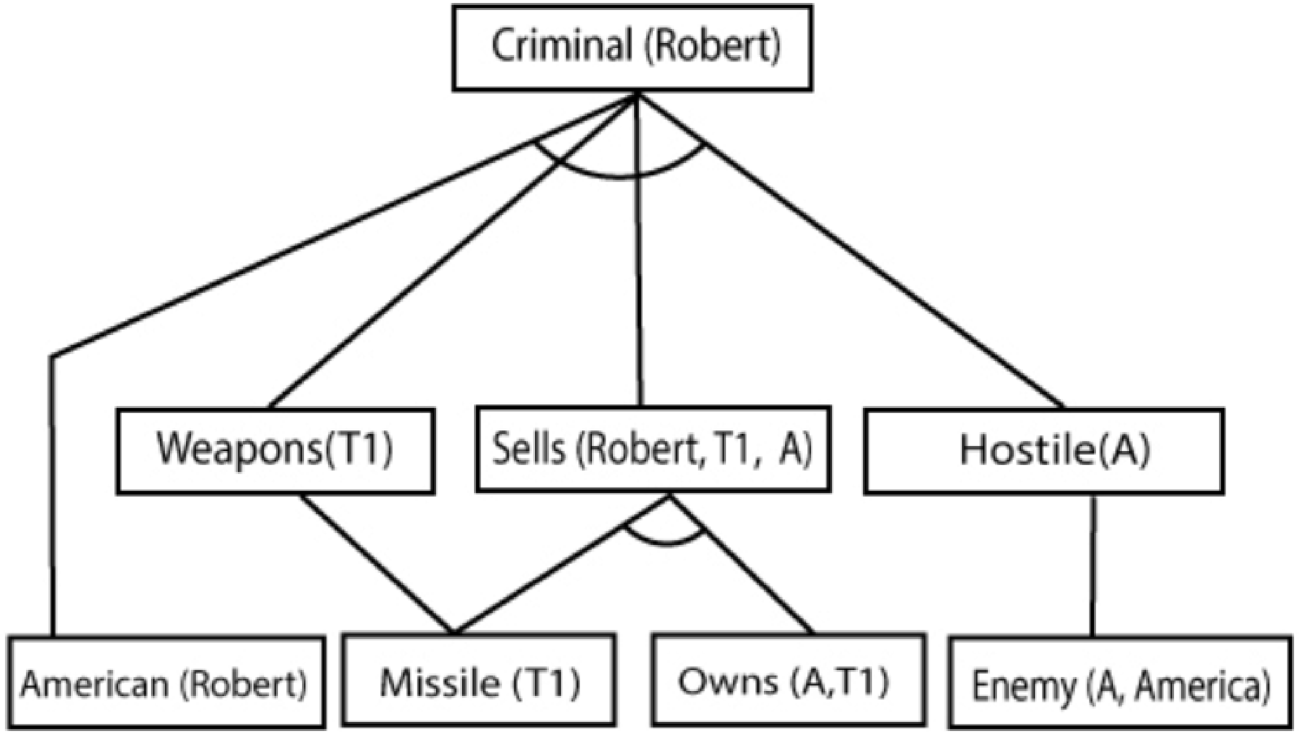
✓Rule-(6) is satisfied with the substitution(p/A), so Hostile(A) is added and which infers from Rule-(7).



FORWARD CHAINING PROOF

- 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.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.
- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

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

- It is known as a top-down approach.
- Backward-chaining is based on modus ponens inference rule.
- 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.
- Backward -chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.
- The backward-chaining method mostly used a depth-first search strategy for proof.

BACKWARD CHAINING: 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 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.

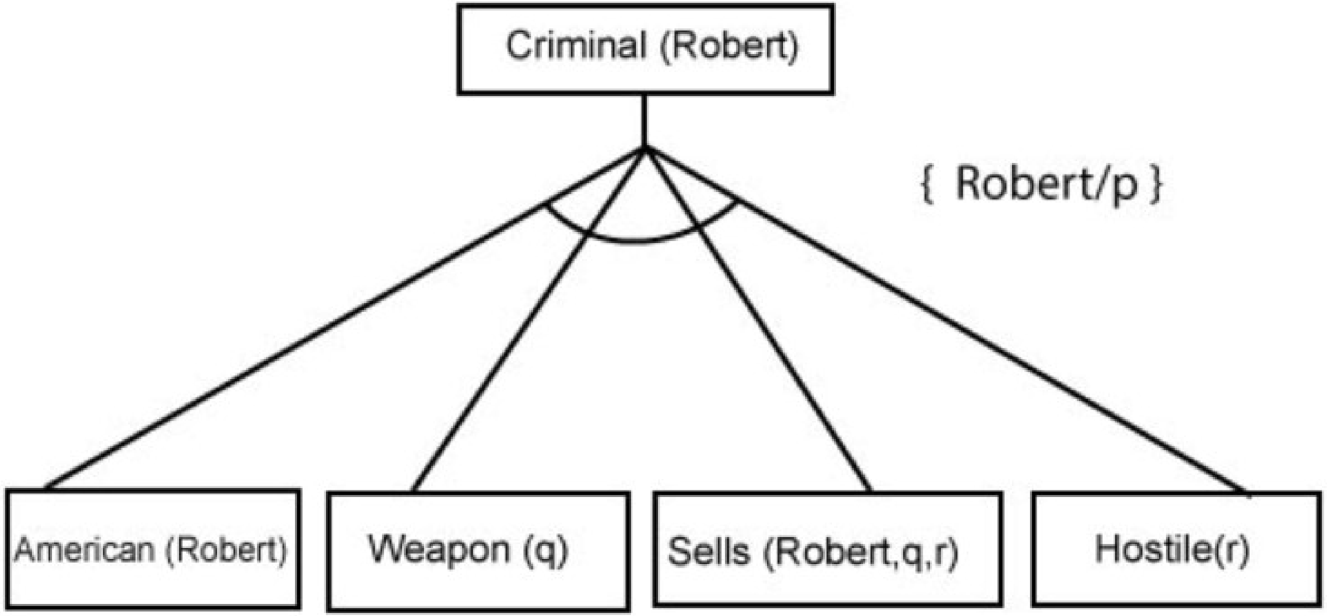
$\boxed{\text{Criminal (Robert)}}$

BACKWARD-CHAINING PROOF

- 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.

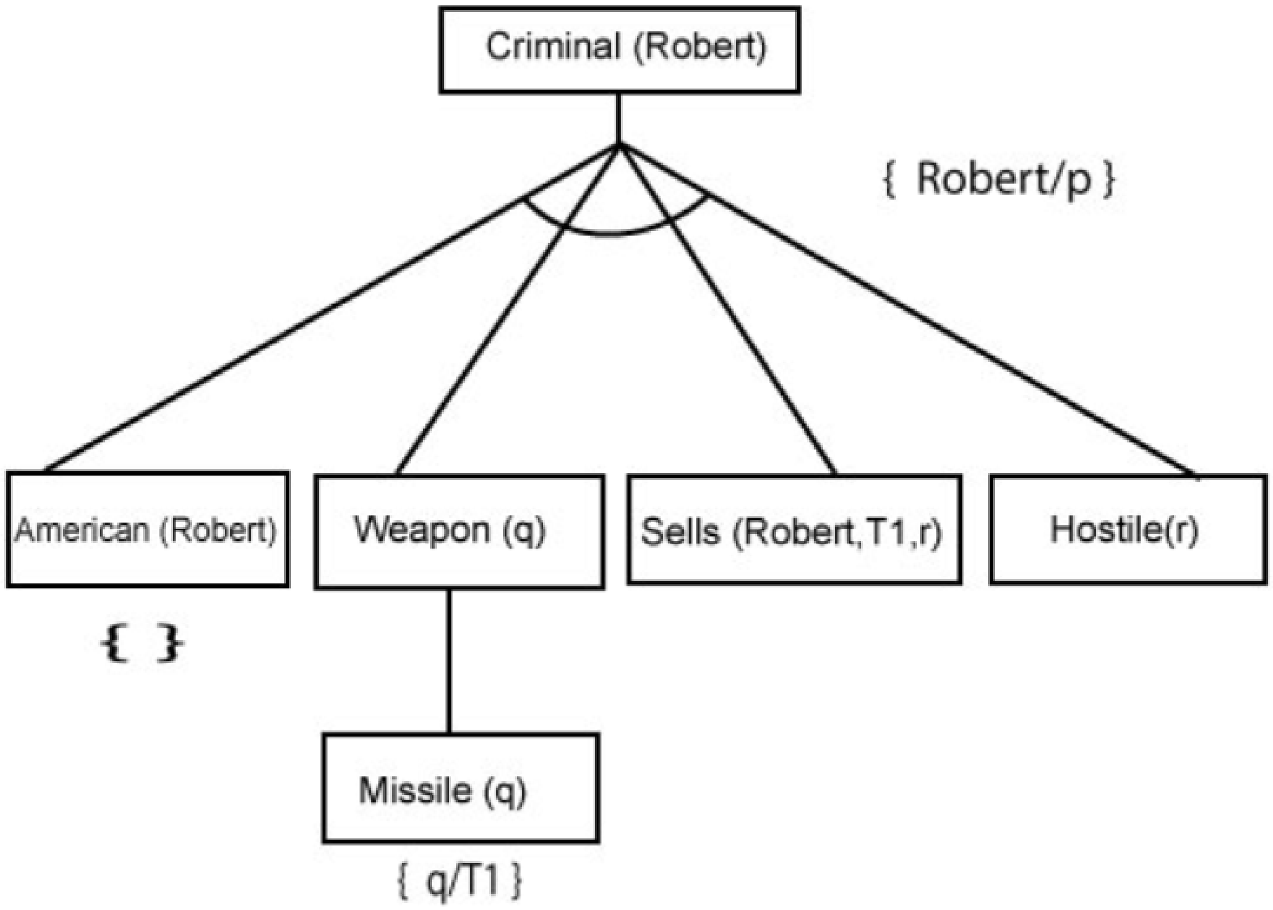
✓ Here we can see American (Robert) is a fact, so it is proved here.



BACKWARD-CHAINING PROOF

- Step-3:

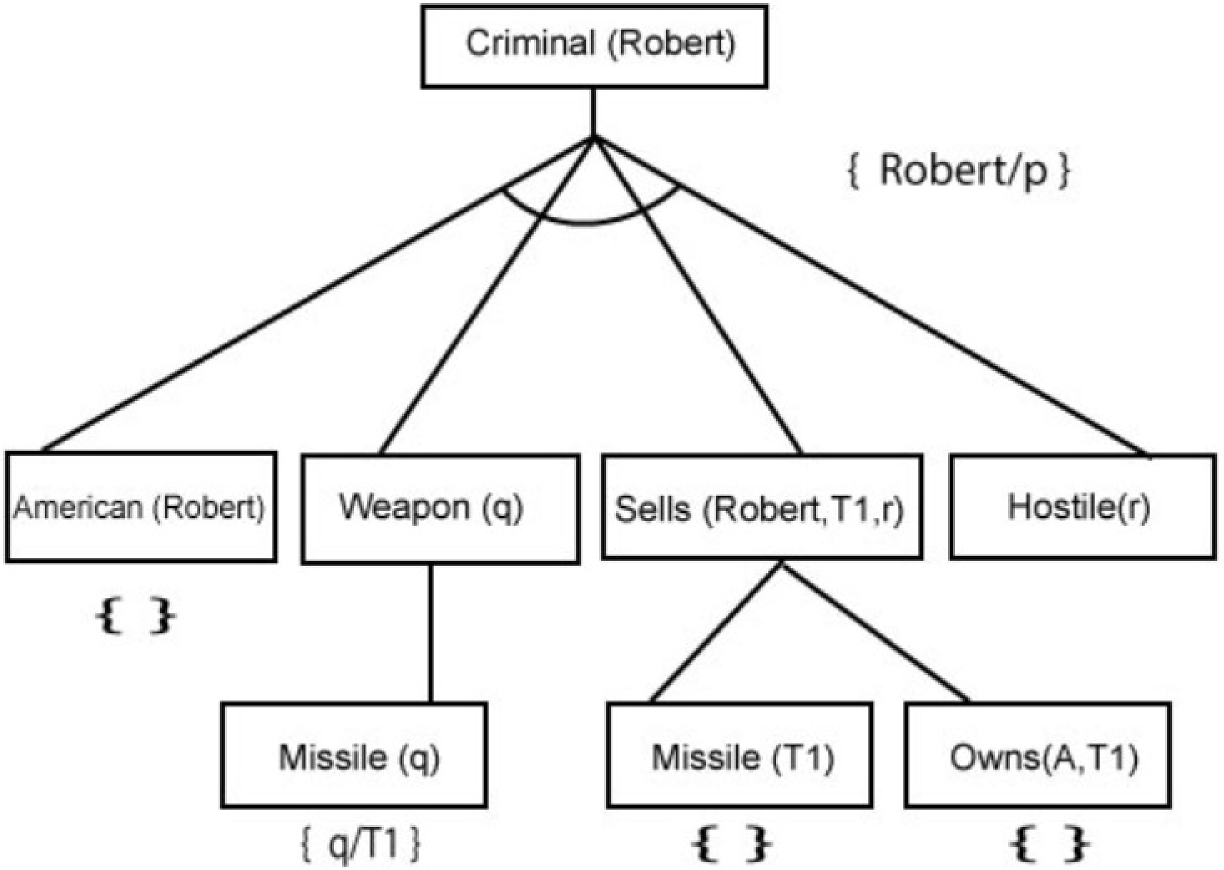
✓ At step-3, we will extract further fact Missile(q) which infer from Weapon(q), as it satisfies Rule-(5). Weapon (q) is also true with the substitution of a constant T1 at q.



BACKWARD-CHAINING PROOF

- Step-4:

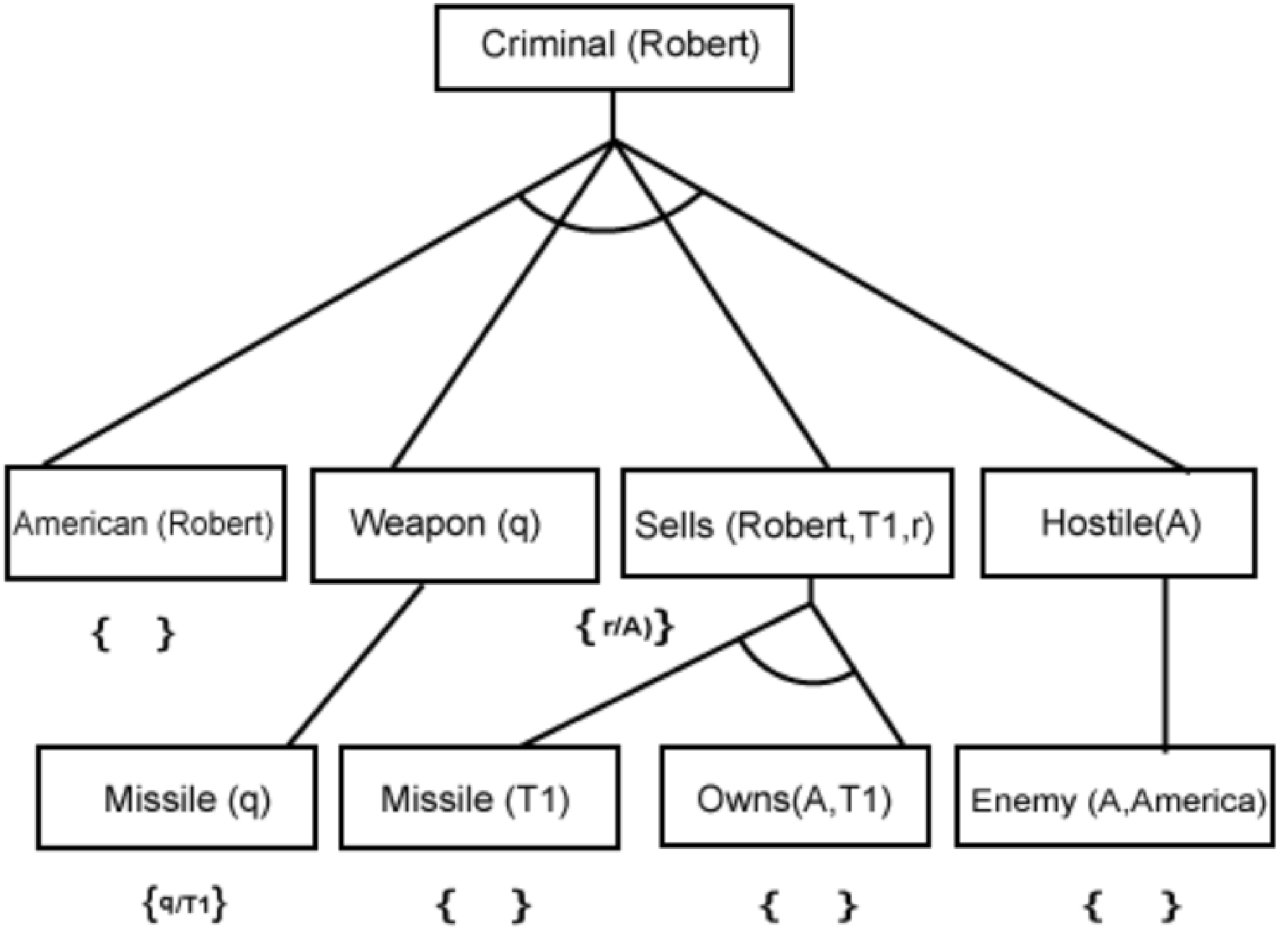
✓ At step-4, we can infer facts Missile(T1) and Owns(A, T1) form Sells(Robert, T1, r) which satisfies the Rule- 4, with the substitution of A in place of r. So these two statements are proved here.



BACKWARD-CHAINING PROOF

- 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.



DIFFERENCE BETWEEN BACKWARD CHAINING AND FORWARD CHAINING

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.

DIFFERENCE BETWEEN BACKWARD CHAINING AND FORWARD CHAINING

GITAM (DEEMED TO BE UNIVERSITY) VISAKHAPATNAM • HYDERABAD • BENGALURU

S. No.	Forward Chaining	Backward Chaining
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.

MODULE IV: ROADMAP

- **First-Order Logic:**
  - ✓ Representation Revisited,
  - ✓ Syntax and Semantics of First-Order Logic,
  - ✓ Using First-Order Logic.
- **Inference in First-Order Logic:**
  - ✓ Propositional vs. First-Order Inference,
  - ✓ Unification and Lifting,
  - ✓ Forward Chaining,
  - ✓ Backward Chaining,
  - ✓ Resolution.

RESOLUTION IN FOL

- Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.
- Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the conjunctive normal form or clausal form.
- Clause: Disjunction of literals (an atomic sentence) is called a clause. It is also known as a unit clause.
- Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be conjunctive normal form or CNF.

STEPS FOR RESOLUTION:

- Conversion of facts into first-order logic.
- Convert FOL statements into CNF
- Negate the statement which needs to prove (proof by contradiction)
- Draw resolution graph (unification).

References:

Text Book(s)

1. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Pearson Publications, 2020

2. Dr. Nilakshi Jain, Artificial Intelligence : Making a System Intelligent, Wiley Publications,1st Edition,2019

Reference Book(s)

1. Elaine Rich, Kevin Knight and Shivashankar B. Nair, Artificial Intelligence, TMH Education Pvt. Ltd., 2008.

2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, Pearson.

Internet:

https://www.javatpoint.com/artificial-intelligence-tutorial

https://www.geeksforgeeks.org/types-of-environments-in-ai/

https://www.tutorialspoint.com/artificial_intelligence/

etc