Module 3

- [x] Knowledge base
- [x] Knowledge based agents
- [x] Propositional logic
- [x] Wumpus World
- [ ] KB for wumpus world
- [ ] First Order Logic
- [ ] Forward chaining
- [ ] Backward chaining

Module 3 -

Logical Agents :

Agents with some representation of complex knowledge about the world / environment & uses inference to derive new information from the knowledge combined with new inputs.

Knowledge Base :

Set of ~~langua~~ sentences in a formal language representing facts about the world

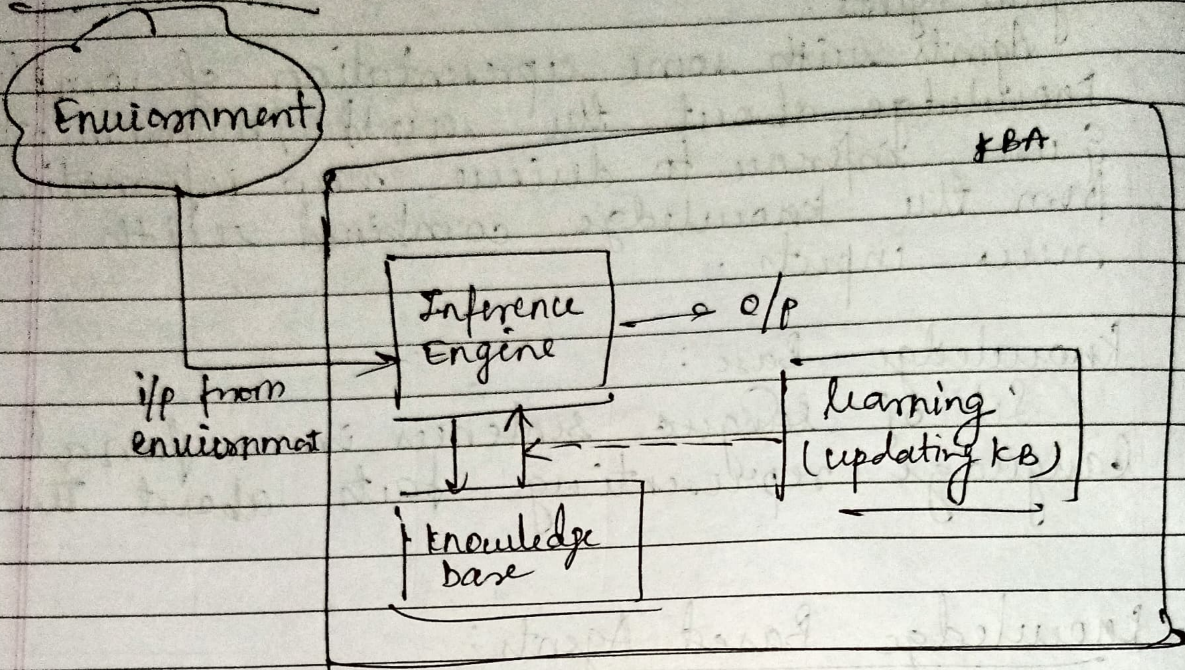
Knowledge Based Agents :

- Intelligent agents need knowledge about the world to make actions / decisions
- ~~Domain specific content~~
- Knowledge Based agents composed of -

    a) Knowledge Base - domain specific content

    b) inference mechanism - domain independent algorithm

Architecture:



~~Typ Approaches~~ Working:

input - percept (perceives the environment)
output - ~~act~~ action
agent maintain KB & initially has some backgr-
-ound knowledge about the real world.

PROPOSITIONAL LOGIC

- Simplest form of logic where all the statements are made by propositions
- A proposition is a declarative statement which is either true or false

Atomic propositions — Compound proposition

Logical Connectives :

- Used to connect two simpler propositions
- Can create compound propositions with the help of logical connectives.
- There are 5 connectives

CONNECTIVE	WORD	TECHNICAL TERM	EXAMPLE
$\wedge$	AND	conjunction	$A \wedge B$
$\vee$	OR	disjunction	$A \vee B$
$\rightarrow$	implies	implication	$A \rightarrow B$
$\leftrightarrow$	if & only if	biconditional	$A \leftrightarrow B$
$\sim$	Not	negation	$\sim A$, $\sim B$

TRUTH TABLES :

For Negation :

P	~P
T	F
F	T

For conjunction:

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

For disjunction

P	Q	P ∨ Q
T	T	T
T	F	T
F	T	T
F	F	F

For implication

P	Q	P → Q
T	T	T
T	F	F
F	T	T
F	F	T

For biconditional

P	Q	P ↔ Q
T	T	T
T	F	F
F	T	F
F	F	T

Precedence :

1 — Paranthesis ( )
2 — Negation (~)
3 — AND (∧)
4 — OR (∨)
5 — implication (→)
6 — bicond (↔)

$A \rightarrow B = \sim A \vee B$

$A \leftrightarrow B = (\sim A \vee B) \wedge (\sim B \vee A)$

Properties of operators :

1) commutativity :

$P \land Q = Q \land P$

2) Associativity

$(P \land Q) \land R = P \land (Q \land R)$

3) Identity element :

$P \land True = P$

~~$P \lor False = False$~~

$P \lor True = True$

$P \land False = False$

$P \lor False = P$

4) Distributive

$P \land (Q \lor R) = (P \land Q) \lor (P \land R)$

5) De-Morgans law

$\sim(P \lor Q) = (\sim P) \land (\sim Q)$

6) Double Negation Elimination

$\sim(\sim P) = P$

Terminologies for inference Rules

implication : $P \to Q$

converse : $Q \to P$

inverse : $\sim P \to \sim Q$

contrapositive : $\sim Q \to \sim P$

Limitations of Propositional logic (PL) :

- We cannot represent all, some, none with PL
- Has limited expressive power
- Cannot describe statement in terms of their properties or logical relationships.

INFERENCE RULES :

Inference : generating conclusions from evidence & facts

Proof : Sequence of conclusion that leads to a desired goal.

- Inference Rules are the templates for generating valid arguments. They are applied to derive proofs in AI.

implication	:	$P \rightarrow Q$	logically equal
converse	:	$Q \rightarrow P$	
inverse	:	$\sim P \rightarrow \sim Q$	
contrapositive	:	$\sim Q \rightarrow \sim P$	

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$\sim P \rightarrow \sim Q$	$\sim Q \rightarrow \sim P$
T	T	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	F	T	T	T	T

Types of Inference Rules

1) Modus Ponens :

If : P & $P \rightarrow Q$ are TRUE
then : Q is TRUE

$$\frac{P \rightarrow Q,\ P}{Q}$$

2) Modus Tollens :
If : $P \rightarrow Q$ & $\sim Q$ are TRUE (i.e. Q is false)
then : $\sim P$ is TRUE (i.e. P is ~~is~~ false)

$$\frac{P \rightarrow Q,\ \sim Q}{\sim P}$$

3) Hypothetical Syllogism :
If : $P \rightarrow R$ & $P \rightarrow Q$ are TRUE
then : $Q \rightarrow R$ is TRUE

$$\frac{P \rightarrow R,\ P \rightarrow Q}{Q \rightarrow R}$$

4) Disjunctive Syllogism :
If : $P \vee Q$ & $\sim P$ is true
then : Q is TRUE

$$\frac{P \vee Q,\ \sim P}{Q}$$

5) Modus Ponens $P \rightarrow Q,\ P\ ;\ Q$
Modus Tollens $P \rightarrow Q, \sim Q\ ,\ \sim P$
Hypothetical Syllogism $P \rightarrow R,\ R \rightarrow Q,\ P \rightarrow Q$
Disjunctive Syllogism $P \vee Q\ ,\ \sim P\ ;\ Q$
Addition $P\ ;\ P \vee Q$
Simplification $P \vee Q\ ;\ P\ ;\ Q$ or $P \vee Q\ ;\ Q$
Resolution $P \vee R\ ,\ \sim P \vee Q\ ,\ Q \vee R$

5) **Addition:**

If : ~~P,Q P⊕Q P~~ ~~TRUE~~ P is TRUE

then : $P \vee Q$ is TRUE

$$\frac{P}{P \vee Q}$$

6) **Simplification:**

If : $P \wedge Q$ is TRUE

then(1) : Q is TRUE

~~then~~(2) : P is TRUE

$$\frac{P \wedge Q}{P} \quad \& \quad \frac{P \wedge Q}{Q}$$

7) **Resolution:**

If : $P \vee Q$ & $\sim P \wedge R$ = TRUE

then : $Q \vee R$ => TRUE

$$\frac{P \vee Q, \sim P \wedge R}{Q \vee R}$$

$$\frac{P \vee Q, \sim P \wedge R}{Q \vee R}$$

KNOWLEDGE BASE FOR WUMPUS WORLD

(1,4)	(2,4)	(3,4)	(4,4)
(1,3)	(2,3)	(3,3)	(4,3)
(1,2)	(2,2)	(3,2)	(4,2)
(1,1) → V	(2,1) B V	(3,1)	(4,1)

[i,j] is used to represent rooms.
i = row no. ; j = column no.

Atomic proposition variable for Wumpus World:

7
- Let P_{ij} be true if there is a pit in room [i,j]
- " B_{ij} " agent perceives breeze "
- W_{ij} " there is a wumpus "
- " $S_{i,j}$ perceives stench "
- " V_{ij} if square [i,j] is visited
- " $G_{i,j}$ if there is gold in "
- $OK_{i,j}$ if the room is safe

∵ the grid is 4×4 and there 7 atomic propositional variable
∴ there are total of 7×4×4 = 122 propositional variables

PROPOSITIONAL RULES FOR THE WUMPUS WORLD:

(R1) $\sim S_{11} \rightarrow (\sim W_{11}) \land (\sim W_{12}) \land (\sim W_{21})$

*(R3) $\sim S_{12} \rightarrow (\sim W_{12}) \land (\sim W_{11}) \land (\sim W_{22}) \land (\sim W_{13})$

*(R2) $\sim S_{21} \rightarrow (\sim W_{21}) \land (\sim W_{11}) \land (\sim W_{22}) \land (\sim W_{31})$

(R4) $S_{12} \rightarrow W_{12} \lor W_{11} \lor W_{22} \lor W_{13}$

Prove that the Wumpus is in (1, 3)

1) ~~Att~~ Apply Modus Ponens (MP) ~~is~~ with $\sim S_{11}$ & R_1:

$\boxed{R_1: \sim S_{11} \rightarrow \sim W_{11} \land \sim W_{12} \land \sim W_{21}}$ $\boxed{\sim S_{11}}$

$\boxed{\sim W_{11} \land \sim W_{12} \land \sim W_{21}}$

we get:

$\sim W_{11} \land \sim W_{12} \land \sim W_{21}$

2) Apply AND-elimination to this, we get 3 sentences:

$\sim W_{11}$,

$\sim W_{12}$,

$\sim W_{21}$

3) Apply MP with $\sim S_{21}$ & R_2:

$\boxed{R_2: \sim S_{21} \rightarrow \sim N_{21} \land \sim W_{22} \land \sim W_{31}}$ $\boxed{\sim S_{21}}$

$\boxed{\sim W_{21} \land \sim W_{22} \land \sim W_{31}}$

we get:

$\sim W_{21} \land \sim W_{22} \land \sim W_{31}$

$\sim S_{11} \rightarrow \ldots (\sim W_{12}) \wedge (\sim W_{27})$

$\sim S_{12} \rightarrow (\sim W_{12}) \wedge (\sim W_{11}) \wedge (\sim W_{22}) \wedge (\sim W_{13})$

$S_{12} \rightarrow W_{12} \vee$

4) Apply AND elimination to this, we get 3 sentences:

$\sim W_{21}$,

$\sim W_{22}$,

$\sim W_{31}$

5) Apply MP to S_{12} & R_4:

$\boxed{R_4: \; S_{12} \rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}}$ $\qquad \boxed{S_{12}}$

$\boxed{W_{13} \vee W_{12} \vee W_{22} \vee W_{11}}$

6) Apply ~~un~~ unit resolution on

$\underset{R_3}{W_{13} \vee W_{12} \vee W_{22} \vee W_{11}}$ & $\sim W_{11}$

we get:

$W_{13} \vee W_{12} \vee W_{22}$

7) Apply unit resolution on

$W_{13} \vee W_{12} \vee W_{22}$ & $\sim W_{22}$

we get:

$W_{13} \vee W_{12}$

8) Apply unit resolution on

$W_{13} \vee W_{12}$ & $\sim W_{12}$

we get:

W_{13}

Proved.

~~PROPOSITIONAL RULES & FOR THE WUMPUS WORLD:~~

~~(R1) $\sim S_{11} \rightarrow (\sim W_{11}) \wedge (\sim W_{12}) \wedge (\sim W_{21})$~~

~~(R2) $\sim S_{12} \rightarrow (\sim W_{12}) \wedge (\sim W_{11}) \wedge (\sim$~~

FIRST ORDER LOGIC:

Syntax:

CONSTANT	1, 2, A, John, mumbai, cat, ...
VARIABLES	x, y, z, a, b
PREDICATES	Brother, Father, >, ...
FUNCTIONS	sqrt, LeftLegOf, ...
CONNECTIVES	$\wedge$, $\vee$, $\sim$, $\rightarrow$, $\leftrightarrow$
EQUALITY	==
QUANTIFIER	$\forall$, $\exists$

First order logic statements can be divided into 2 parts:

Subject: Main part of the sentence

Predicate: Relation that binds two atoms together in a statement.

Eg:

$\underbrace{x}_{\text{subj}}$ $\underbrace{\text{is an integer}}_{\text{predicate}}$

predicate

Representation: $G(x)$ subject, $\equiv$ x is an integer

QUANTIFIERS :

- Quantifiers are words that refer to quantities such as "some" & "or" "all".
- It tells you for how many elements a given predicate is true.

Universal   Existential

UNIVERSAL :

- The Universal quantification of $P(x)$ is the statement
- "$P(x)$ for all values of x in the domain"
- Notation : $\forall x \; P(x)$
  - ↳ for all x, $P(x)$

EXISTENTIAL :

- The existential quantification of $P(x)$ is the proposition ~~"@th~~
  "there exists an element x in the domain such that $P(x)$."
- $\exists x \; P(x)$
  - ↳ for some x; $P(x)$

The main connective of $\forall$ is '$\rightarrow$'
$\exists$ is '$\wedge$'

Q ~

INFERENCE IN FIRST ~~LOGIC~~ ORDER LOGIC :

Terminologies :

SUBSTITUTION :

- fundamental operation performed on terms & formulas.
- F [a/x] means substitute constant 'a' ~~inst~~ in 'x'

EQUALITY :

- We use equality symbols which specify that the two terms refer to the same object.

~~Exp~~ Eg: $\sim(x = y)$ is equivalent to $x \neq y$

FOL inference rules :

1) Universal Generalization :

- P(c) is true for any arbitrary element c in the domain, then we can have ~~to~~ a conclusion as $\forall x P(x)$

$$\frac{P(c)}{\forall x \, P(x)}$$

- This rule can be used to show that every element has similar property.

universal generalization $P(c) \vdash \forall x\, P(x)$
universal instantiation $\forall x\, P(x) \vdash P(c)$
existential generalization $P(c) \vdash \exists x\, P(x)$
existential instantiation $\exists x\, P(x) \vdash P(c)$

2) Universal Instantiation:

- We can infer any sentence obtained by substituting a ground term for the variable.
- We can infer $P(c)$ by substituting c from the domain from $\forall x\, P(x)$

$$\frac{\forall x\, P(x)}{P(c)}$$

- Can be applied multiple times
- Previous KB is logically equivalent to the new KB.

3) Existential Instantiation:

- The rule states that one can infer $P(c)$ from the formula given in the form of $\exists x\, P(x)$ for a new constant 'c' for which $P(c)$ is true

- $$\frac{\exists x\, P(x)}{P(c)}$$

- Can only be applied once
- The new KB is not logically equivalent to old KB.

4) Existential Introduction

- This rule states that if there is some element c in the domain that has the property P, then we can infer that there exists something in the universe which has the property P

$$\frac{P(c)}{\exists x\, P(x)}$$

UNIFICATION :

- Process of making two different logical atomic expressions identical by finding a substitution.
- Let φ_1 & φ_2 be two atomic sentences & σ be a unifier, such that,

$$\varphi_1\sigma = \varphi_2\sigma$$

then it can be expressed as UNIFY (φ_1, φ_2)

Example:

1) Find Most General Unifier (MGU) for
UNIFY {King(x), King(John)}
Substitution : θ = {John/x}

2) for UNIFY {P(x,y) & P(a, f(z))}
substition = $\frac{a}{x}$, $\frac{f(z)}{y}$

Conditions:

1) Predicate symbol must be same.
2) No. of arguments in both expressions must be identical.
3) No two similar variables should be present in the same expression

RESOLUTION

~~Example~~ Example:

Example:

a. John likes all kinds of food $\forall x: food(x) \Rightarrow likes(John, x)$

b. Apple & vegetable are food

c. Anything anyone eats & not get killed is food

d. Anil eats ~~everyt~~ peanuts and still alive

e. Harry eats ~~everyt~~ everything that Anil eats

f. John likes peanuts (T.P.T.)

STEP 1: Conversion of facts into FOL

a. $\forall x: food(x) \longrightarrow likes(John, x)$

b. $food(Apple) \wedge food(vegetable)$

c. $\forall x \forall y: [eats(x, y) \wedge \sim killed(x)] \longrightarrow food(y)$

d. $eats(Anil, peanuts) \wedge alive(Anil)$

e. $\forall x: eats(Anil, x) \rightarrow eats(Harry, x)$

f. $\forall x$:

f. h. $likes(John, peanuts)$ (T.P.T)

g. f. $\forall x: \sim killed(x) \longrightarrow alive(x)$ — Added Predicates

h. g. h $\forall x: alive \longrightarrow \sim killed(x)$

STEP 2: Conversion of FOL to CNF

a. $\sim food(x) \vee likes(John, x)$

b. $food(Apple) \wedge food(vegetable)$ — can be written as 2 different statements → b_1. food(Apple) b_2. food(vegetable)

c. $\sim eats(y, z) \vee killed(y) \vee food(z)$

d. $eats(Anil, peanuts) \wedge alive(Anil)$ — [d_1. eats(Anil, peanuts) d_2. alive(Anil)]

e. $\sim eats(Anil, w) \vee eats(Harry, w)$

g. $killed(g) \vee alive(g)$

h. g. ~~killed(k) V ~alive(g)~~ $\sim alive(k) \vee \sim killed(k)$

f. h. $likes(John, peanuts)$

Rewriting all the CNF :

1) ~food(x) v likes(John, x)
2) food(Apple)
3) food(Vegetable)
4) ~eats(y, z) v killed(y) v food(z)
5) eats(Anil, ~~Pe~~ Peanuts)
6) alive(Anil)
7) ~eats(Anil, w) v eats(Harry, w)
8) killed(g) v alive(g)
9) ~alive(k) v ~killed(k)
10) likes(John, Peanuts)

DRAW RESOLUTION GRAPH :

~likes(John, Peanuts) ~food(x) v likes(John,

(# Peanuts/x)

~food(Peanuts) ~eats(y, z) v killed(y) v food

(# Peanuts/z)

~eats(y, Peanuts) v killed(y) eats(Anil, Peanuts)

(# Anil/y)

killed(Anil) ~alive(k) v ~killed(

(# Anil/k)

~alive(Anil) alive(Anil)

$\square$

The negation of the conclusion has been proved as a complete contradiction with the given set of statements.

FORWARD CHAINING

i) American (p) ∧ Weapon (q) ∧ Sells (p, q, r) ∧ hostile (r) ⟶ criminal (p)

ii) ∀p: Owns (A, p) ∧ Missile (p)
   ii. a) Owns (A, T₁)
   ii. b) Missiple (T₁)
} Existenial instation, T₁ is a constant

iii) ∀p: Missiles (p) ∧ Owns (A, p) ⟶ Sells (Robert, p, A)

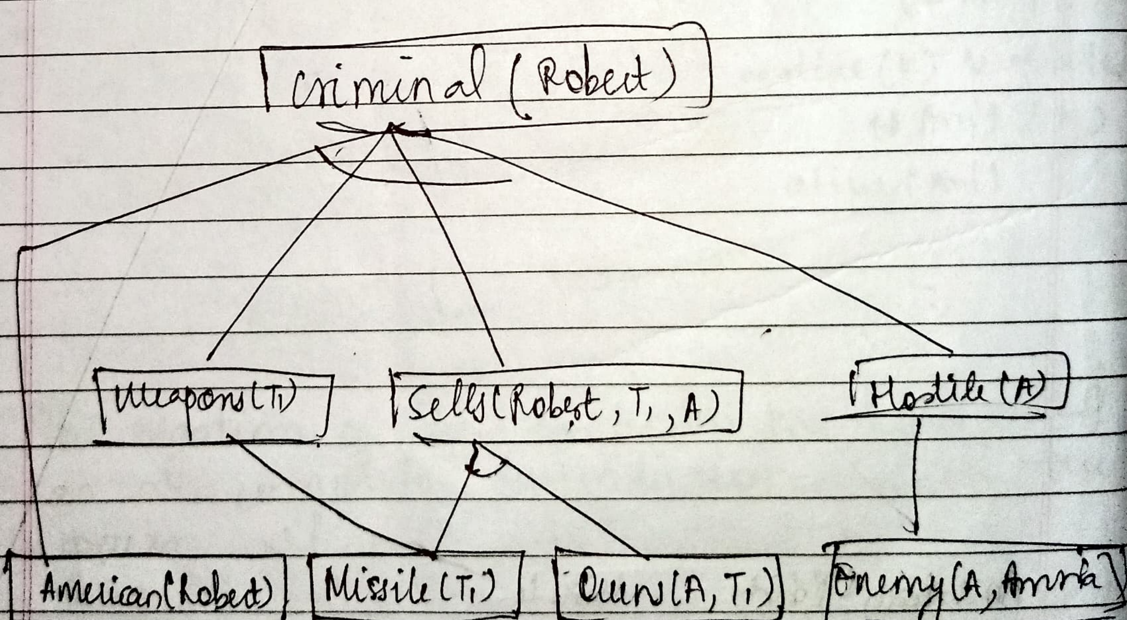
iv) Missile (p) ⟶ Weapons (p)

v) Enemy (p, America) ⟶ Hostile (p)

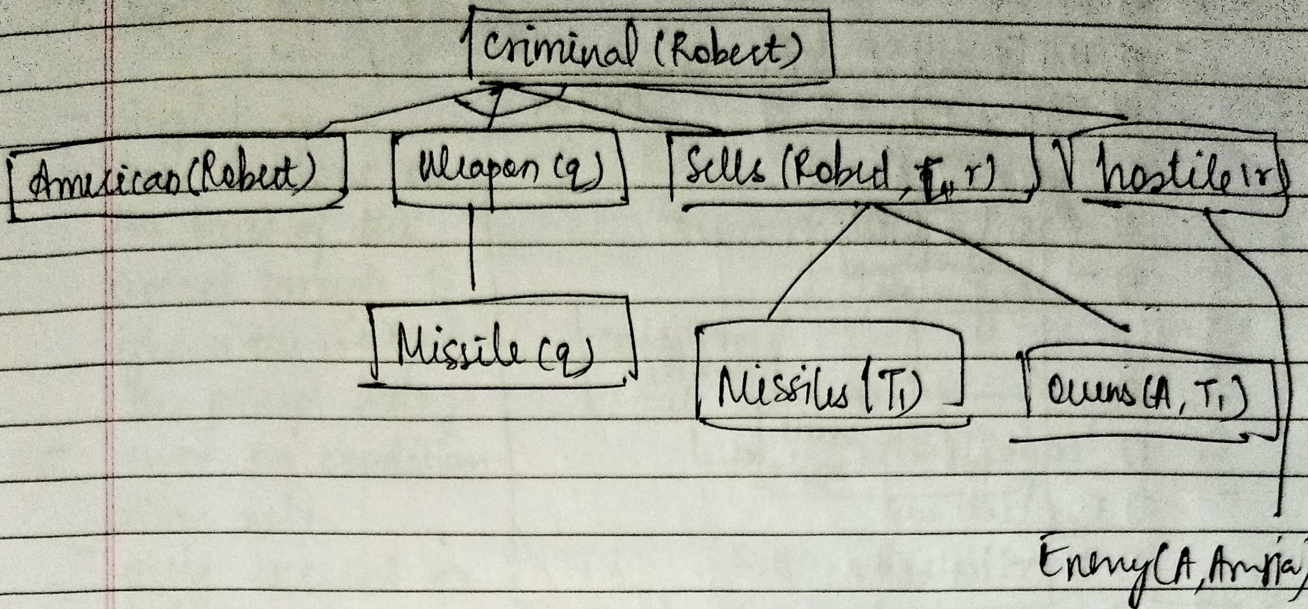
vi) Enemy (A, America)

vii) American (Robert)

FORWARD CHAINING



BACKWARD CHAINING



Forward chaining	Backward chaining
atomic sent in KB - inference rules - extracts data until goal is reached	- goal node - works backward to find known facts that support the goal.
- down up approach	- top down
- process of making a conclusion out of the known facts	- process of ~~proving~~ finding facts & prove the goal true.
- goal node is not known	- goal node is known
- data driven	- Goal driven.
- BFS	- DFS
- commonly used in expert sys	- commonly used in game theory, inference engine