R

$P \rightarrow (Q \vee \neg R)$

Propositional Calculus

It is a system that deals with the method used for manipulation of the symbols according to some rules

Alphabet Set:

i) Set of variables or propositional symbols P, Q, R

ii) Logical constants → True (T)
→ False (F)

iii) Two parantheses "(" and ")"

iv) Set of logical operators

	word	symbol	example
a)	not	$\neg$	$\neg X$
b)	and	$\wedge$	$X \wedge Y$
c)	or	$\vee$	$X \vee Y$
d)	implies	$\rightarrow$	$(X \rightarrow Y)$ if X then Y
e)	if and only if	$\leftrightarrow$	$(X \leftrightarrow Y)$

Evergreen
Page No.
Date / /20

Examples x: It is hot
y: It is humid
z: It is raining

(1) if it is $\underbrace{\text{humid}}_{(Y)}$ then it is $\underbrace{\text{hot}}_{(X)}$

$\hookrightarrow \boxed{(Y \rightarrow X)}$

(2) if it is $\underbrace{\text{hot}}_{X}$ and $\underbrace{\text{humid}}_{Y}$ then it is not $\underbrace{\text{raining}}_{Z}$

$\boxed{(X \wedge Y) \rightarrow \neg Z}$

$\Rightarrow$ {note: Propositional logic is not applicable in case of All, some}

v) Set of equivalence relations or laws:-
(P, Q, R) are variables

Commutative law: $P \wedge Q \cong Q \wedge P$, $P \vee Q \cong Q \vee P$

Associative law: $(P \wedge Q) \wedge R \cong P \wedge (Q \wedge R)$,
$(P \vee Q) \vee R \cong P \vee (Q \vee R)$

Double Negation: $\sim(\sim P) \cong P$

De-Morgan's law: $\neg(P \vee Q) = \neg P \wedge \neg Q$,

$\neg(P \wedge Q) \cong \neg P \vee \neg Q$

Evergreen
Page No.
Date: / /20

Absorption law: $p \wedge (p \vee q) \equiv p$
$p \vee (p \wedge q) \equiv p$

Law of contradiction: $p \wedge \neg p \equiv$ False

Law of excluded middle: $p \vee \neg p \equiv$ True

Law of impotency: $p \wedge p \equiv p$

Rules of Inference:

① Modus Ponens: if 'p' and '$p \to q$' is given to be true, then we can infer that 'q' is true

ex

P: it is a holiday — (T)
Q: The school is closed] → we can infer that it is true
$P \to Q$: if it is holiday then the school is closed — (T)

② Modus Tollens: if '~~¬q~~ q' and '$\sim p \to \sim q$' are given to be true, then we can infer that $\sim p$ is true

$\neg q$ = School is not closed
$(\sim p \to \sim q)$ = if it is not a holiday then school is not closed.
↓
$\sim p$ → It is not a holiday (True)

Tautology and Truth Table

Truth Table shows how the truth or falsity of a compound statement depends on the truth or falsity of simple statements.

Some of Truth Tables :-

① Negation

P	~P
T	F
F	T

② AND : $(P \wedge Q)$ → True when P and Q both are true

simple statements   compound statement

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

Tautology → It's a formula which is always true

Contradiction → It is a formula which is always false (opposite of tautology)

Q Prove that $(P \to Q) \to (P \to Q) \vee (Q \to P)$ is a tautology

P	Q	P → Q	Q → P	(P→Q) ∨ (Q→P)
T	T	T	T	T
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

all are true

~~ex Suppose it's true that you get an A~~

ex If you get an A, then I'll give you a dollar
The statement will be true if I keep my promise and false if I don't

① Suppose if it's true that you get an A and it's true that I give you a dollar. Since I kept my promise, implication is true. ~~this correspond~~

② Suppose it's true that you get an A but it's false that I give you a dollar. Since I did not keep my promise, the implication is false.

③ What if it's false that you get an A? whether or not I give you a dollar

~~Representing~~

Evergreen
Page No.
Date: / /20

First order Predicate Logic (FOPL)

Representing simple facts in FOPL:

↳ It models world in terms of objects

Real world facts can be represented as logical propositions written as well formed formulas in propositional logic

Symbols — ① objects
② properties
③ Relation
} Representation

Symbols are formed of the following :-
① Set of all uppercase english alphabets
② Set of digits from 0 to 9
③ underscore character

[All dogs are brown] can't be written in propositional logic
↳ for such types of sentences, First order Predicate Logic (FOPL) is used

→ Every atomic sentence is a sentence. It is defined as predicate constant of arity 'n', followed by t1, t2, --- tn terms ~~ent~~ enclosed in parantheses and separated by commas.
① if 'S' is a sentence then ¬S is sentence.
② if S1, S2 are sentences, then S1 ∧ S2 is also a sentence (conjunction)
③ if S1, S2 are sentences, then S1 V S2 is also a sentence (disjunction)

Evergreen
Page No.
Date : / /20

(4) if S1, S2 are sentences, $S1 \rightarrow S2$ is also a sentence (implication)

(5) if S1, S2 are sentences, $S1 \equiv S2$ ~~is~~ also a sentence (equivalence)

(6) if X is a variable and S is sentence then $\forall X$, S is a sentence

(7) if X is a variable and S is a sentence then $\exists X$ S is a sentence

Quantifiers in Predicate Calculus:-

There are two quantifiers used in First order predicate ~~la~~ calculus

(1) Universal] for all 'x' such that represented as $\forall x$

(2) Existential] for some 'x' such that $\exists x$

→ Quantifier constraint the meaning of sentence containing a variable
→ Quantifier is followed by a variable and a sentence

Example

(1) All boys like football
(2) Some boys like football

Evergreen
Page No.
Date: / /20

(1) $\forall x$: Boys(x) $\rightarrow$ Like(x, football)

~~(2) $\exists x$: Boys Boys(x) $\wedge$ (L~~

(2) $\exists x$: Boys(x) $\wedge$ Like(x, football)

ie if x is a boy, he may like football

(3) Gorilla is black
~~first wor~~ In FOPL, words are first represented with symbols

Gorilla(x) $\rightarrow$ Black(x)

~~(4) Every person who buys a policy is smart~~

(4) All boys like cricket (like (verb))
$\forall x$: boys(x) $\rightarrow$ like(boys, cricket) (we need to find action word (verb))
: boys(x) $\rightarrow$ like(x, cricket)

(5) Some boys like football
$\exists x$: boys(x) $\wedge$ like (x, football)
(Incase of Some, $\wedge$ is used)

(6) Some girls hate football
$\exists x$: girls(x) $\wedge$ hate(x, football)

(7) All girls love pink $\forall x$: girls(x) $\rightarrow$ love(x, Pink)

(8) Every person who buys a ~~plo~~ policy is smart
$\forall x \forall y$: Person(x) $\wedge$ Policy(y) ~~buys(x, y)~~ $\wedge$ buys(x,y) $\rightarrow$ Smart(x)

(9) No person buys expensive policy
$\forall x \forall y$: Person(x) $\wedge$ Policy(y) $\wedge$ expensive(y) $\rightarrow$ $\neg$buys(x,y)
not($\neg$)

Knowledge Representation

In order to solve complex problems in AI, we need
1) ~~larget~~ large amount of knowledge
2) mechanism to manipulate knowledge to create solution to new problems.

* Methods used to represent knowledge:—
1) Propositional Logic
2) Predicate ~~ka~~ Logic
3) Semantic network
4) Frame

Propositional Logic

Proposition is a statement of a fact.

eg:- Shyam is an honest boy } Propositional logic statement

Is Shyam an honest boy? (not a PL)

$7 \times 5 + 4 \neq 48$ → False (F)

Any statement that is either ~~or~~ 1 true or false is called a proposition
↳ ie any declarative sentence is a proposition

~~Fro~~

Propositional symbols / variables
A, B, C , P, Q, R — — —

ex Shyam is an honest boy. Shyam is hardworking

$\underbrace{\text{Shyam is an honest boy}}_{p}$. $\underbrace{\text{Shyam is hardworking}}_{q}$

Simple Statements

Shyam is an honest boy + Shyam is hardworking

↳ composite statement
Shyam is an honest boy and hardworking
$\wedge$
conjuctive

$\Rightarrow$ $p \wedge q$

Connectives
- $\wedge$ and
- $\vee$ or
- $\rightarrow$ implies (conditional)
- $\leftrightarrow$ (Biconditional), if and only if
- $\neg$ not

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$\neg p$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

if I is T
II is F

if I & II are same then ans → T otherwise F

T

Limitations

1. No symbol for All
2. No symbol for there exist

To overcome this issue, First order Predicate logic (FOPL) is used

Predicate Logic

Predicate logic is an extension of propositional logic. It allows the structure of fact and sentence to be defined.

Ex 1) the ball's colour is red

In preposition logic
a = the Ball's colour is Red

In Predicate logic
Color (Ball, Red) → arguments / objects
↓
Predicate / Relation

Ex 2) Rohan likes Bananas
like (Rohan, Bananas)

Ex All students are intelligent
Rohan is a student
↳ infrence : Rohan is intelligent

Limitations of Proposition logic
1) Such types of infrences can't be drawn in proposition logic
So there was a need of language that could involve object, relation and ~~property~~ property.

→ Predicate logic involved these concepts so it was better knowledge representation language than proposition logic

① → There are some sentences that can't be represented in propositional logic
↳ It has symbols $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
there is no symbol for Every(all) & some.

→ Predicate logic makes the use of quantifiers for ~~sentences containing~~ All & Some

e.g. Everybody wants to play some game

↳ can't be expressed using proposition logic but can be expressed using predicate logic

Quantifiers

we need quantifiers to express english words including All & Some

① Universal $\forall$ (for all)
② Existential $\exists$ (there exist (some))

e.g. Everybody wants to play some game
all → x
Predicate → L
y

$\forall x \exists y \; L(x, y)$ = Play(Everybody, someone)

Represent Facts or Statements into well formed formula in Predicate Logic

Example (1)

1. Marcus was a man.
2. Marcus was a pompeian.
3. Marcus was born in 40 A.D.
4. All men are mortal.
5. All pompeians died when the volcano erupted in 79 A.D
6. No mortal lives longer than 150 years
7. It is now 1991.
8. Alive means not dead.
9. if someone dies, then he is dead at all later times

(1) Marcus was a man
man(Marcus)

(2) Marcus was a pompeian
Pompeian (Marcus)

(3) ~~bo~~ Marcus was born in 40 A.D
born (Marcus, 40)

(4) All men are mortal
$\forall x : man(x) \rightarrow mortal(x)$

(5) All pompeians died when volcano erupted in 79 A.D.
$\forall x : Pompeian(x) \rightarrow died(x, 79)$
erupted (volcano, 79)

(6) No mortal lives longer than 150 years
let us assume mortal has born at time t1 and current time be t2, now $t2 > t1$

Evergreen
Page No.
Date: / /20

$\forall x : \forall t_1 : \forall t_2$: mortal $(x) \wedge$ born (x, t_1)
$\wedge\ gt(t_2 - t_1, 150) \rightarrow dead(x, t_2)$

(7) It is now 1991
now = 1991

(8) Alive means not dead
$\forall x : \forall t$: [alive $(x, t) \rightarrow \neg$ dead (x, t) $\wedge$
[$\neg$ dead $(x, t) \rightarrow$ alive (x, t)]

(9) if someone dies, then he is dead at all later times
$\forall x : \forall t_1 : \forall t_2$: died $(x, t_1) \wedge gt(t_2, t_1) \rightarrow$ ~~dead~~
dead(x, t_2)

e.g. $t_1 = 1900$ $t_2 = 1901$
$t_2 > t_1$ if $t_2 = 1899$ then it is not true

x is dead at t_1 means x is dead at t_2 also

example (2)
1. Marcus was a man man(Marcus)
2. Marcus was pompeian pompeian (Marcus)
3. All pompeian were Romans
$\forall x$: Pompeian $(x) \rightarrow$ Roman(x)
4. Caesar was a ruler
ruler(Caesar)
5. All romans were either loyal to caesar or hated him
$\forall x$: Roman $(x) \rightarrow$ loyalto $(x$, Caesar$) \vee$ hate$(x$, caesar$)$
(if x is a roman then x is loyal to caesar or x hates caesar)

either of the two possibilities will exist

Evergreen
Page No.
Date: / /20

6. Everyone is loyal to someone
$\underbrace{\text{Everyone}}_{x}$ is loyal to $\underbrace{\text{someone}}_{y}$

$\forall x : \exists y : \text{loyalto}(x, y)$

7. People only try to assassinate rulers they are not loyal to

$\forall x : \forall y : \text{person}(x) \wedge \text{ruler}(y) \wedge \text{tryassissnate}(x,y) \rightarrow \neg \text{loyalto}(x, y)$

8. Marcus tried to assassinate caesar.
$\underbrace{\text{Marcus}}_{x}$ tried to assassinate $\underbrace{\text{caesar}}_{y}$.

$\text{tryassassinate}(\text{Marcus}, \text{Caesar})$

9. All men are ~~people~~ person
$\forall x : \text{man}(x) \rightarrow \text{person}(x)$

Prooving infrences from facts

Step ① Convert statements (facts) into the logical statements
Step ② Use these logical statements to find the infrence

Ex ① was Marcus loyal to Caesar?

nil

$\downarrow (1)$

man(Marcus)

$\downarrow (9)$

person(Marcus)

$\downarrow (8)$

person(Marcus) $\wedge$ tryassassinate(Marcus, Caesar)

$\downarrow (4)$

person(Marcus) $\wedge$ tryassissnate(Marcus, Caesar) $\wedge$ ruler(Caesar)

Evergreen
Page No.
Date: / /20

$\downarrow$ (7, substitution)

$\neg$ loyalto (Marcus, Caesar)

(i.e Marcus is not loyal to Caesar)

(2) Was Marcus hates Caesar?

Marcus is not loyal to caesar

to prove Now in part (1), we have already proved that Marcus is not loyal to Caesar

we need to prove

$\neg$ loyalto (Marcus, Caesar)

$\downarrow$ (2)

Pompeian (Marcus)

$\neg$ loyalto (Marcus, Caesar)

$\downarrow$ (3)

Roman (Marcus)

$\neg$ loyalto (Marcus, Caesar)

$\downarrow$ (5)

hate (Marcus, caesar)

Propositional logic VS First order Logic (Predicate logic)

Propositional logic	First order Logic
1. It uses preposition in which complete sentence is denoted by symbol ie it takes a sentence and represents it with a symbol, e.g It is cold P: It is cold $\neg P$: It is not cold	1. FOL uses predicate which involves constants, variables, functions and relations. Here we make a relation function of a particular sentence and predict

2. PL cannot represent individual entities
e.g Meera is short
PL cannot find relation for the sentence if meera is short then she cannot climb the wall.

P : Meera is short
q : She cannot climb the wall

P → q (ie if P then q)
It can be used for such types of sentences. we can't find relation of a simple sentence.

2. FOL can represent individual properties
Eg Short (Meera)
Predicate constant

3. It cannot express Generalization, Specialization or pattern.
Quantifiers are not used PL.
Eg Triangles have 3 sides

3. It can express generalization, specialization or pattern
Eg No.-of-sides (triangle, 3)
Predicate (No.of sides of triangle is 3)

(Imp: It is not necessary to have predicate in the sentence, we can define our own predicate and arguments of the predicate should be in a correct order otherwise its interpretation will be wrong.

Evergreen
Page No.
Date : / /20

Resolution in AI

The sentences can be converted in FOL and on the basis of FOL, ~~we so~~ something can proved by using the facts of FOL

Here we need to make a predicate that can be used commonly in all facts in order to prove whatever is given

eg ① a) Ravi likes all kind of food
b) Apple and chicken are food.
c) Anything anyone eats and is not killed is food.
d) Ajay eats peanuts and still alive
e) Rita eats that Ajay eats.
Anyone who is not killed is ~~not~~ alive
g) Prove Ravi likes Peanuts
Solve on the basis of above facts given

" Ravi likes Peanuts "

likes (Ravi , Peanuts)

Hint : check for likes in the given facts

Steps for Resolution (To prove resolution)
→ Negate the statement to be proved (Prove contradict)
→ Convert given facts into FOL (First order Logic)
→ Convert FOL into CNF (Conjunctive Normal Form / closed Normal Form)

→ Draw resolution graph

Step ① Convert into FOL

a) Ravi likes all kind of food.

$\forall_x$ food(x) $\longrightarrow$ like(Ravi, x)

b) Apple and chicken are food

i) food(apple) ii) food(chicken)

d) Ajay eats peanuts and still alive

eats(ajay, peanuts) $\wedge$ alive(ajay)

Note: Here we are considering (d) first as this fact will be used in part (c)

c) ~~Anything~~ ~~anyone~~ anyone eats and is not killed is food

$\forall_x \forall_y$: eats(x, y) $\wedge$ $\neg$ killed(x) $\longrightarrow$ food(y)

(Note order of predicate should be same throughout

$\forall_x$: killed(x) $\longrightarrow$ $\neg$alive(x)

$\forall_x$ alive(x) $\longrightarrow$ $\neg$killed(x)

$\Rightarrow$ # Rules to convert FOL into CNF

~~rename variable~~

① Eliminate '$\rightarrow$' and '$\leftrightarrow$'

$a \rightarrow b$ $\therefore \neg a \vee b$

$a \leftrightarrow b$ $a \rightarrow b \wedge b \rightarrow a$

② Move '$\neg$' inward

$\rightarrow$ $\neg(\forall_x p) = \exists_x \neg p$

$\rightarrow$ $\neg(\exists_x p) = \forall_x \neg p$

$\rightarrow$ $\neg(a \vee b) = \neg a \wedge \neg b$

$\rightarrow$ $\neg(a \wedge b) = \neg a \vee \neg b$

$\rightarrow$ $\neg\neg a = a$

Evergreen
Page No.
Date / /20

③ Rename variable
④ Replace Existential quantifier by skolem constant
$\exists x\ Rich(x) = Rich(G_1)$
⑤ Drop universal quantifier

Convert FOL into CNF

a) continuing with eg ①

a) $\forall x\ food(x) \rightarrow likes(Ravi, x)$
$\neg food(x) \vee likes(Ravi, x)$

d) $\forall x \forall y : eats(x, y) \wedge \neg killed(x) \rightarrow food(y)$

$\rightarrow \neg [eats(x, y) \wedge \neg killed(x)] \vee food(y)$
$\rightarrow \neg eats(x, y) \vee killed(x) \vee food(y)$

c) (i) will remain as it is
$eats(ajay, peanuts) \wedge alive(ajay)$
$\forall x\ \neg killed(x) \rightarrow alive(x) \Rightarrow killed(x) \vee alive(x)$
$\forall x : alive(x) \rightarrow \neg killed(x) \rightarrow \neg alive(x) \vee \neg killed(x)$

CNF
i) $\neg food(x) \vee likes(Ravi, x)$
ii) $\neg eats(x, y) \vee killed(x) \vee food(y)$
iii) $eats(ajay, peanuts) \wedge alive(ajay)$
iv) $killed(x) \vee alive(x)$
v) $\neg alive(x) \vee \neg killed(x)$

This types of sentences with $\wedge$ can be converted into individual sentences
$eats(ajay, peanuts)$ / $alive(ajay)$

Resolution Graph

start with contradiction

$\neg$ likes (Ravi, peanuts)

x/peanuts (i) food

$\neg$ food (peanuts)

y/peanuts ii)

$\neg$ eats (x, peanuts) $\vee$ killed (x)

x/Ajay iii)

killed (Ajay)

x/Ajay iv)

$\neg$ alive (ajay) alive (ajay)

ϕ

---

b) $\forall x : \overset{a}{\text{killed}}(x) \rightarrow \neg \overset{b}{\text{alive}}(x)$

g)

~~⇒~~ $\neg$ killed ~~x~~ $\neg$ alive(x)

$\neg$ alived $\vee$ $\neg$ killed(x)

Evergreen
Page No.
Date / /20

$\forall$ ~~¬ likes (John, Peanuts)~~ ~~¬~~ food(Peanuts) $\vee$ ~~likes (John, Peanuts)~~

$\forall x$ Course(Science, Hard)

$\forall x$ Course(x); course(Science, x) $\Rightarrow$ hard(x)

Infrence Engine

It is a component of the system applies logical rules to the knowledge base to deduce new information.

Forward Chaining / Forward reasoning

→ Starts with the ~~new~~ known facts and asserts new facts.

Backward chaining / Backward reasoning

→ Starts with goals and works backward to determine what facts must be asserted so that goals can be achieved

Forward Eg Conclude from "A" and "A → B" to "B"

A: It is raining

B: If it is raining then the road is wet

Forward chaining: From A and A → B, we can conclude that the road is wet

Backward chaining

Conclude from "B" & "A → B" to "A"

B: The road is wet
If it is raining, the road is wet
(A → B)
A → It is raining

~~Backward chaining~~

Forward chaining

→ when based on the available data a decision is taken then the process is called as the forward chaining.
→ It works from initial state and work towards goal state.

Example Given facts:

① → It is a crime for an American to sell weapons to the enemy of America
② – Country Nono is an enemy of America
③ – Nono has some missiles
④ – All the missiles were sold to Nono by
⑤ colonel.
⑥ – Missile is a weapon
⑦ – Colonel is an American

"we have to prove that colonel is a criminal

Step ① Convert facts into FOL

① American(x) ∧ weapon(y) ∧ sell(x, y, z) ∧ enemy (z, America) ⇒ Criminal(x)

② Enemy(Nono, America)

③ owns (Nono, x)

④ Missile(x)

⑤ ∀x Missile(x) ∧ owns (Nono, x) ⇒ sell(colonel, x, Nono)

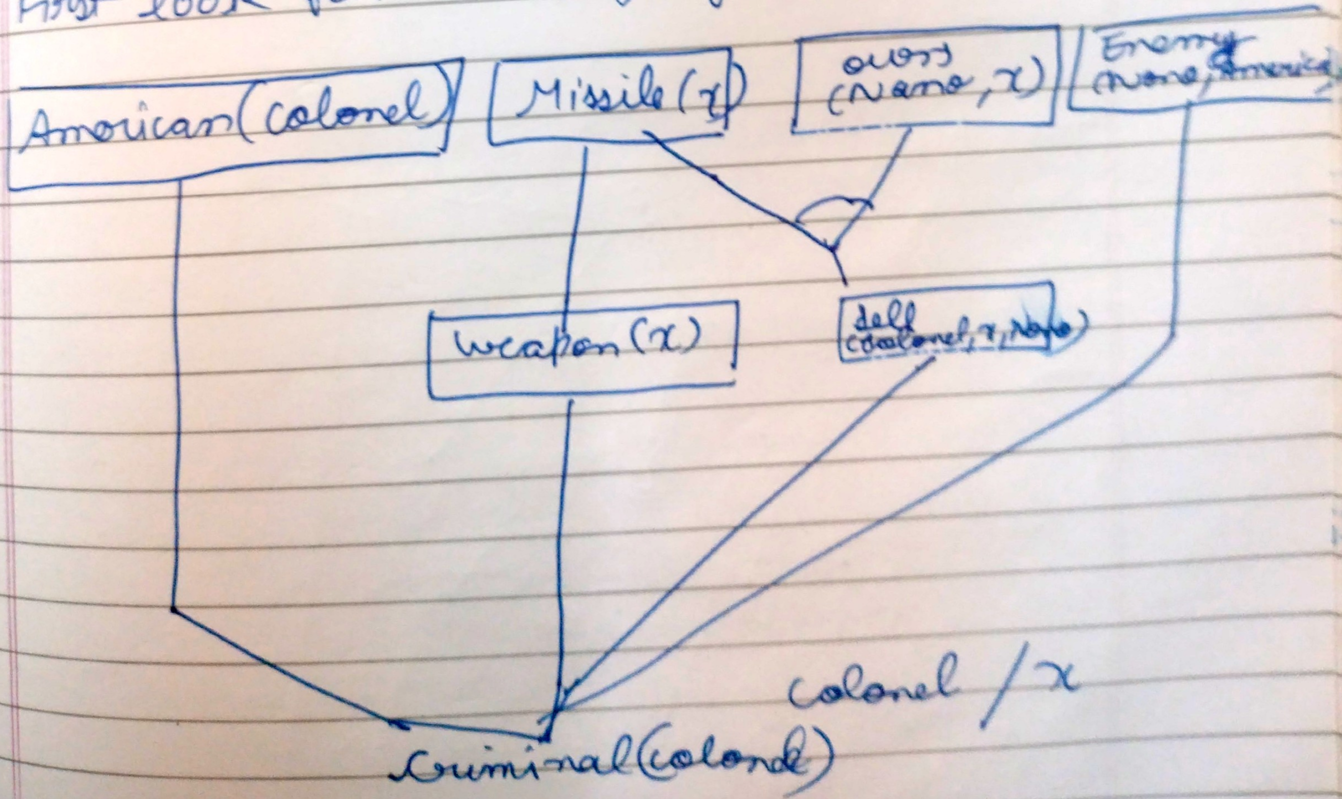
⑥ Missile(x) ⇒ weapon(x)

⑦ American(colonel)

Proof by forward chaining

Start from initial state to goal state

First look for all single facts



Proof by backward chaining