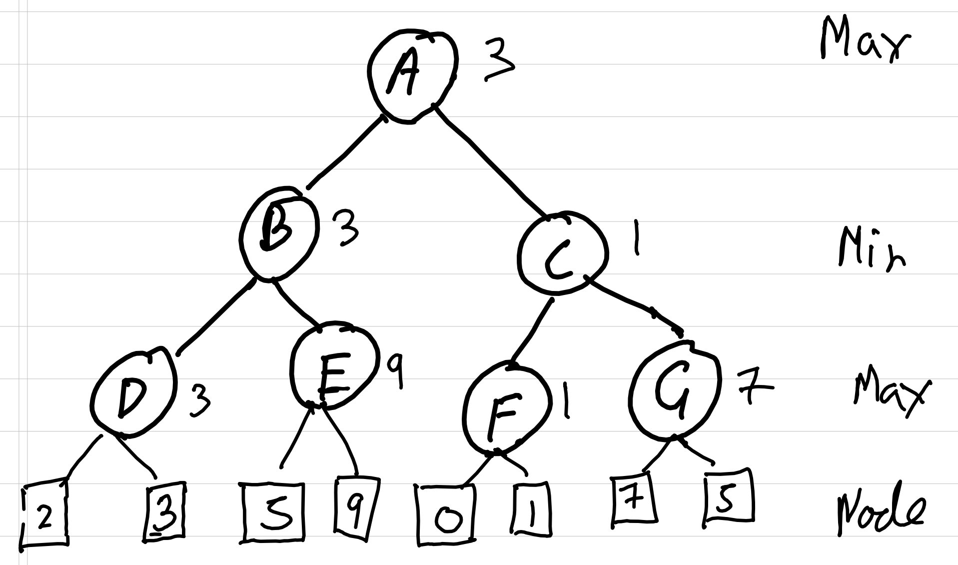
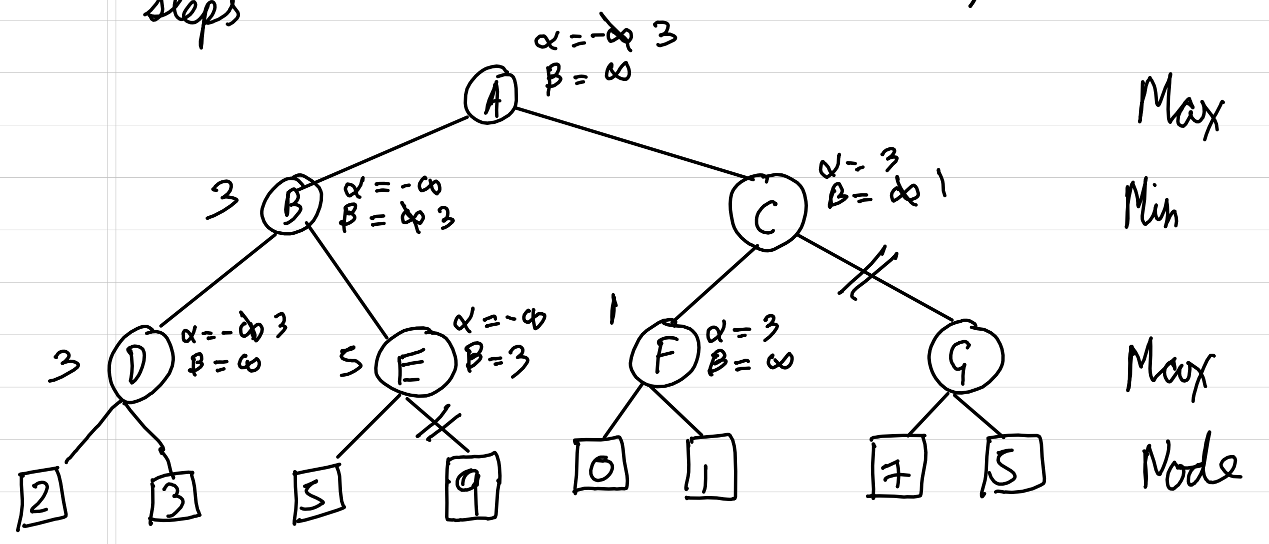
① Min Max → Adverserial search



α-β - Pruning
Used to reduce the number of steps



At D $\alpha = -\infty$
$\beta = \infty$

Comparing 2, $\alpha < 2$ ∴ Update $\alpha = 2$

$\alpha = 2$
$\beta = \infty$
$\alpha \geq \beta \to$ False

Comparing 3 $\alpha < 3$ ∴ Update $\alpha = 3$

$\alpha = 3$
$\beta = \infty$
$\alpha \geq \beta \to$ False

Path to D → 3

At B $\alpha = -\infty$
$\beta = \infty$

Compare D $\beta > 3$ ∴ Update β

$\alpha = -\infty$
$\beta = 3$
$\alpha \geq \beta \to$ false

At E $\alpha = -\infty$
$\beta = 3$
} Passed from parent
$\alpha \geq \beta \to$ false

Compare 5

$\alpha < 5$ ∴ Update 5

$\alpha = 5$
$\beta = 3$
$\alpha \geq \beta \to$ True

∴ Prune branch to 9
Cost to E = 5

At B Backtracking →

$\beta < 5$ ∴ No Update
Cost to B = 3

At A Backtracking

$\alpha < 3$ ∴ Update $\alpha = 3$

$\alpha = 3$
$\beta = \infty$
$\alpha \geq \beta \to$ False

At C, $\alpha = 3$
$\beta = \infty$
} Pass to child

At F, $\alpha = 3$
$\beta = \infty$
} Pass to child

Compare with 0, $\alpha > 0$. ∴ No Update
Compare with 1, $\alpha > 1$ ∴ No Update

$\alpha \geq \beta \to$ False

Maximum value at F

0, 1 → 1

Backtrack to C

$\beta > 1$ ∴ Update $\beta = 1$

$\alpha = 3$
$\beta = 1$
$\alpha \geq \beta$

∴ Prune

② Adversarial Search

Enemy, opponent changing state of problem in every step

Alternating Min & Max levels (Self & Opponent)

Used to play games like checkers, chess

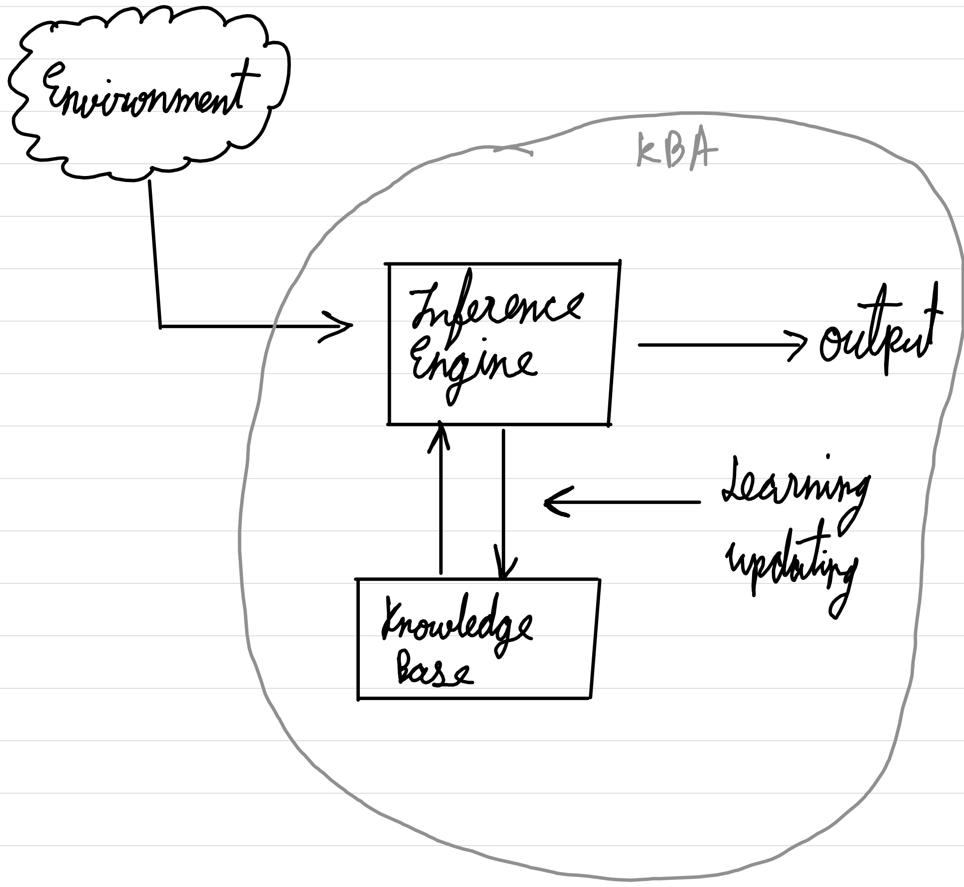
Min Max algorithm → Complexity $O(b^d)$

α-β pruning → Reduce Complexity $O(\sqrt{b^d})$
Prune certain nodes
Condition $\alpha \geq \beta$
$\left.\begin{array}{l}\alpha \rightarrow \text{Max} \\ \beta \rightarrow \text{Min}\end{array}\right\}$ Path from current node to root Node

③ Knowledge Based Agents

→ Agents that rely on knowledge & reasoning to take decisions over simple reflexes

They make deductions, Decisions, Conclusions } Based on knowledge base



Knowledge based agents have the ability to

T — Take actions
R — Reason over knowledge
U — Update knowledge
M — Maintain internal knowledge base

Represent the world with some formal representation and act intelligently

Three operations performed by KBA in order to show intelligent behaviour

T — Tell the knowledge base the situation
A — Ask what to Do
P — Perform the selected Action

④ Wumpus World

P – Navigate the world, kill Wumpus, get gold & return safely

+1000 getting gold
+100 killing a Wumpus

−10 using up arrow
−1000 getting eaten by Wumpus
−1000 falling in pit
−1 every Move taken

Game ends when agent climbs out with gold or when agent dies

E – Deterministic
Single Agent
Static
Discrete
Partially Observable
Known
Sequential

4×4 grid rooms

Random locations for Wumpus & gold & pit

A – Move left
Move right
Move up
Move Down
Shoot arrow Left
Right
up
Down

S – feel Breeze

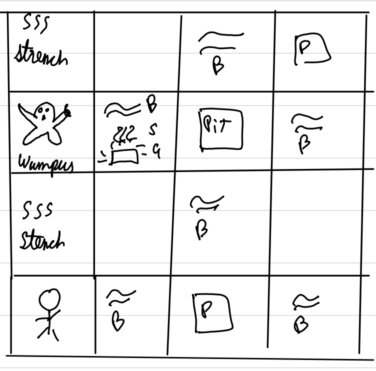
feel strentch

feel glitter

feel Bump (on wall)

feel Scream (when Wumpus is killed)

Prove that Wumpus is in room(1, 3) using FOL



Rules

$\sim S_{11} \rightarrow \sim W_{12} \wedge \sim W_{21}$ ①
$S_{12} \rightarrow W_{11} \vee W_{22} \vee W_{13}$ ②
$\sim S_{21} \rightarrow \sim W_{11} \wedge \sim W_{22} \wedge \sim W_{31}$ ③

from ① Using AND elimination

$\sim S_{11} \rightarrow \sim W_{12}$
$\sim S_{11} \rightarrow \sim W_{21}$
} Apply Modus Ponens $\sim S_{11}$

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

from ③ similarly

$\sim W_{11}$, $\sim W_{12}$ $\sim W_{31}$

from ② Modus Ponens

$W_{11} \vee W_{12} \vee W_{13}$ ④

Unit resolution ④ & W_{11}, W_{12}

$\therefore W_{13}$ obtained

5. Limitations of PL logic

Cannot represent states like ALL, Some or none

Cannot model real world situations

Limited expressive power

Cannot handle uncertainty

⑥ Hill Climbing Problem

Continuously Move in direction of increasing value

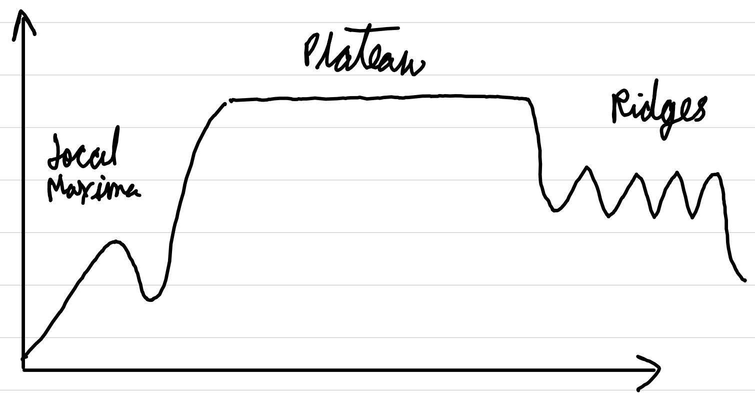
Terminate when peak value is reached

Does not maintain search tree. Only current state & value of objective function

Dont look beyond immediate neighbours

Heuristic local search algorithm

Disadvantages → Local Maxima reached
∴ Cant Navigate ridges, plateaus



Start with a initial solution and take small steps to choose best neighbours

7 Convert to FOL

1	Fred is a colie	Colie (Fred)
2	Sam is Freds Master	Master (Sam, Fred)
3	Day is saturday	Day (saturday)
4	It is cold on Saturday	cold (saturday)
5	Fred is trained	Trained (Fred)
6	Trained coolies are good dogs	$\forall x$ Colie(x) $\wedge$ Trained(x) $\rightarrow$ good dog (x)
7	If a dog is a good dog and has a master at some place then he will be at that place	$\forall x y z$ gooddog (x) $\wedge$ Master(y, x) $\wedge$ At(y, z) $\rightarrow$ At(x, z)
8	If day is saturday & day is cold then sam is at Meuseum	Day (saturday) $\wedge$ Day (cold) $\rightarrow$ At(Sam, Mesum)

⑧ Prove by Resolution

Fred is at Museum

Writing in CNF

$\sim(Trained(x) \wedge Collie(x)) \vee GoodDog(x)$

Demorgan's law

$\equiv \sim Trained(x) \vee \sim Collie(x) \vee GoodDog(x)$ ⑨

Similarly

$\sim Day(saturday) \vee \sim Cold(saturday)$
$\vee At(Sam, Museum)$ ⑩

$\sim GoodDog(i) \vee \sim Master(j, i)$
$\vee \sim At(j, z) \vee At(i, z)$ ⑪

Assume Fred is Not at Museum

⑪ $\sim At(Fred, Museum)$

i | Fred
z | Museum

Resolution $a \vee \sim B, \sim B \therefore a$

$\sim GoodDog(Fred) \vee \sim Master(j, Fred)$
$\vee \sim At(j, Museum)$

⑩

j | sam Resolution

$\sim Day(saturday) \vee \sim Cold(saturday)$
$\vee \sim Gooddog(Fred) \vee \sim Master(sam, Fred)$

③, ④, ②

Resolution

$\sim Gooddog(Fred)$

⑨

x(Fred)

$\sim Trained(Fred) \vee \sim Collie(Fred)$

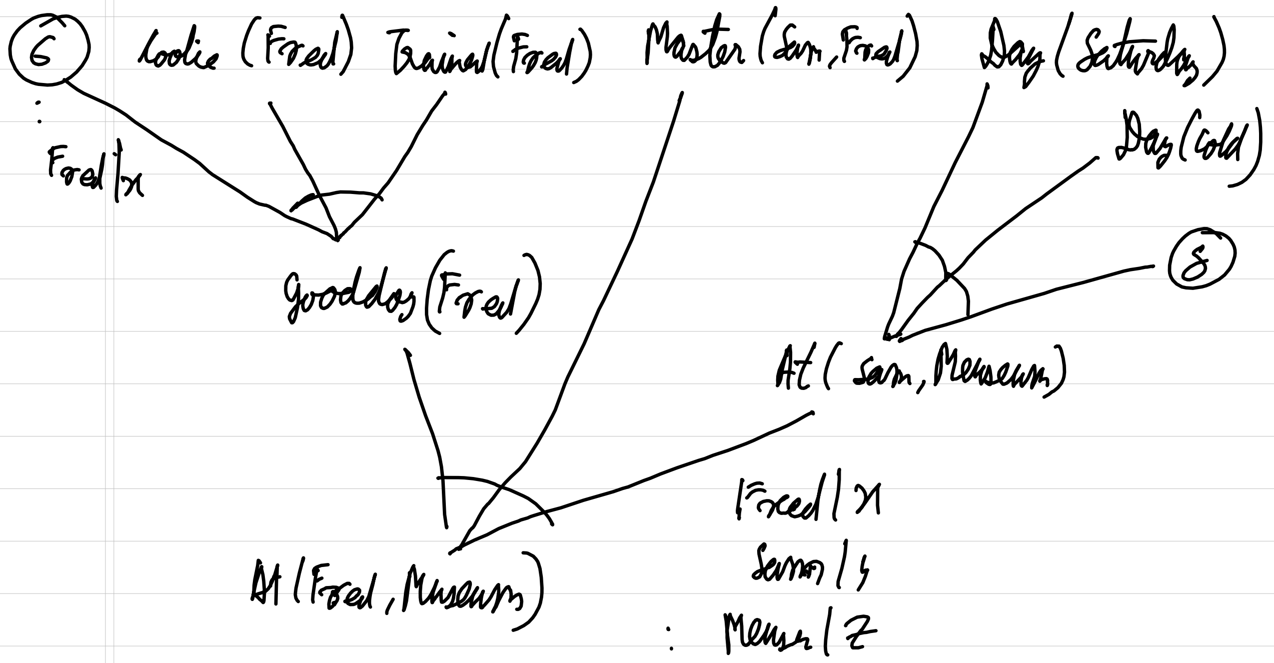
① ⑤

Resolution

F

Hence proved

⑨ Proove by forward / Backward chaining



(10) Forward chaining Vs Backward Chaining

Forward chaining	Backward Chaining
Start with known facts and proceed towards the goal	start from the goal & proceed towards known facts
Bottom up	Top down
Breadth first	Depth first
Any conclusion	Only required data
Planning & control	Diagnosis & debugging
∞ no of conclusions	Only finite conclusion

11. What is Uncertainty in AI & how to solve it?

→ In real world, there are a lot of scenerios where certainty of something is not confirmed.

In order to represent uncertain knowledge where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning

When there are Unpredictable outcomes

too large probabilities

Unknown errors / events

In order to tackel uncertainty, probabilistic reasoning is used.

Probabilistic models represent the relationships between events in an probabilistic way

Then inference uses the probability to take decisions.

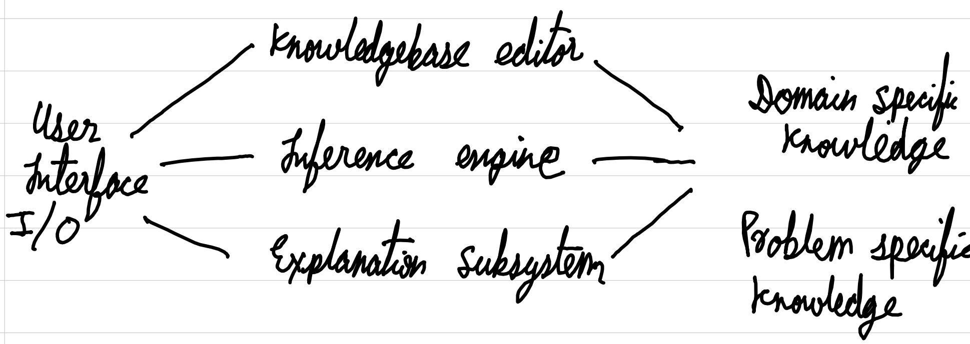
(12) ADL vs STRIPS

ADL	STRIPS
Action description language	Stanford Research Institute Problem Solver
Allows Negative literals	Negative literals not allowed
Open world Assumption Unknown literals are unknown	Closed world assumption Unknown literals are false
$P \wedge \sim Q$ means add $P, \sim Q$ delete $\sim P$ & Q	$P \wedge \sim Q$ means add P delete Q
Goals are conjunctions & disjunctions $A \wedge (B \vee C)$	Only conjunctions is goals $(A \wedge B)$
Type support $x = y$	No type support
equality support P: plane	No equality support

(13) Explain expert systems with diagram

→ Program that implements knowledge & reasoning process of a human expert.

"An intelligent computer program that employes knowledge and inference procedures to solve problems that are considered difficult enough to require significant human expertise for their solutions"



They are efficient, accurate & solve problems like human experts

Limitations of expert systems —

Cant handle unforseen situation
High Cost
Difficult to maintain
No creativity

(14) STRIPS for Air Cargo Problem

Move C_1 from A_1 to A_2
Move C_2 from A_2 to A_1

Init (At (C_1 A_1), At (C_2 A_2),
At (P_1 A_1), At (P_2 A_2),
Cargo (C_1), Cargo (C_2), Airport (A_1)
Plane (P_1), Plane (P_2), Airport (A_2)
)

Goal (At (C_1 A_2), At (C_2 A_1))

Actions (Load (C, P, a)

Precondition : Cargo (c) $\wedge$ Plane (p) $\wedge$ Airport (a) $\wedge$ $\sim$ In (c, P)

Post condition : In (c, P) $\wedge$ $\neg$ At (c, a)

Fly (P, from, to)

Precond : Plane (P) $\wedge$ Airport from $\wedge$ airport to) $\wedge$ at (P, from)

Postcond : $\sim$at (P, from) $\wedge$ at (P, to)

Unload (P, C, a)

Precond : In (C, P) $\wedge$ Cargo (c) $\wedge$ Plane (P) $\wedge$ airport (P)

Post cond : at (c, a) $\wedge$ $\sim$ In (c, P)

15 POP for air cargo

→ Goal : $at(C_1, a_2) \cap at(C_2\ a_1)$

Consider action $Unload(C_1, P_1, a_2)$

Post cond : $at(C_1, a_2) \cap \sim In(C_1, P_1)$

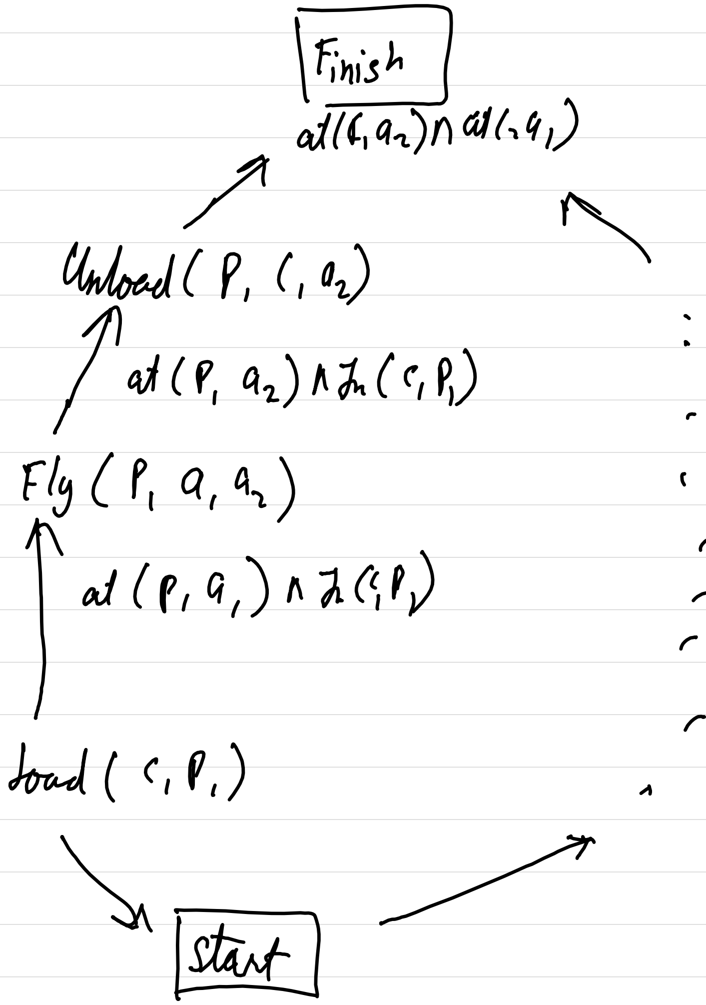
Precond : $In(C, P) \cap at(P, a)$, $\cap$ . . . Type

Consider action $Fly(P_1, a_1, a_2)$

Post cond . . .

Precond - .

⋮



16. POP vs TOP

POP	TOP
Parallel execution of action sequences	sequential execution of action
No specific order of actions	Exact ordering of actions
Single graph Obtained	Multiple orderings obtained

(17) Inference rules in FOL

Universal instantiation

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

if all people are donkeys then c is a donkey

Universal Generalization

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

P(c) is true for any c then $\forall x \; P(x)$

Any cat is a animal → all cats are animals
A byte has 8 bits → all bytes have 8 bits

Exential instantiation

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

There exists an intelligent monkey $\exists x \; P(x)$

let monkey be named C $P(c)$

Exestential Generalization

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

Tinky is an intelligent monkey $P(c)$

∴ There exists an intelligent monkey $\exists x \; P(x)$

(18) Inference rules for knowledge reasoning

If you study you will get good marks

Modus Ponens

$$\frac{P \rightarrow q, \; P}{\therefore q}$$

You studied

∴ You have got good marks

Modus Tollens

(Law of contrapositive)

$$\frac{P \rightarrow q, \; \sim q}{\therefore \sim P}$$

You didnt get good marks

∴ you havent studied

And elimination

$$\frac{a \wedge b}{a}$$

You are smart & cute

∴ you are smart can be infered

Bidirectional elimination

$$\frac{(a \rightarrow b) \wedge (b \rightarrow a)}{a \leftrightarrow b} \qquad \frac{a \leftrightarrow b}{(a \rightarrow b) \wedge b \rightarrow a}$$

Resolution

$$\frac{P \vee q, \; \sim q \vee r}{\therefore P \vee r}$$

Unit resolution

$$\frac{P \vee q, \; \sim q}{\therefore P}$$

19 Process of building expert system

i) Problem identification

ii) Conceptulization

iii) Formalization

iv) Implementation

v) Testing

(20) Steps of NLP

→

1) Segmentation
2) Tokenization
3) Stemming
4) Lemmatization
5) Identifying stop words
6) Dependancy parsing
7) POS tagging
8) Named entity Recognition
9) Chunking

21 Unification & lifting

Unification is the process of making two different logical expressions identical by finding a substitution

King(x), King(John)

x | John

Unify algorithm takes in two atomic sentences and returns a unifier if it exists