- ## Knowledge

→ Knowledge-based agents → agents that reason by operating on internal representations of Knowledge

Ex⇒ P: If it didn't rain ~~today~~, Harry would visit A today.
01 Q: Harry visited A or B today not both.
R: Harry visited B today.

Knowledge → Harry didn't visit A today.
(inference) It's raining today.

→ Sentence → an assertion about the world in a Knowledge representation language

→ Knowledge base → a set of sentences known by a Knowledge-based agent.

→ Entailment ($\alpha \models \beta$) → in every model in which sentence α is true, sentence β is also true.

→ inference → the process of deriving new sentences from old ones.

Ex:02 P: it is a tuesday.
Q: it is raining.
R: Harry will go for a run.

KB: $(P \wedge \neg Q) \rightarrow R$ {(P AND NOT Q) implies R}

inference: R

#Search OO

(BST).

Date
Page

Model Checking:

- To determine if $KB \models \alpha$:
  - Enumerate all possible models.
  - If in every model where KB is true, α is true, then KB entails α.
  - Otherwise, KB does not entail α.

Query: R (Ex.02)

P	Q	R	KB
False	False	False	False
False	False	True	"
False	True	False	"
False	True	True	"
True	False	False	"
True	False	True	True
True	True	False	False
True	True	True	"

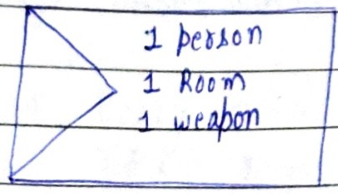
Knowledge Engineering:

The process of trying to take a problem & figure out what propositional symbols to use in order to encode that idea, or how to represent it logically is known as Knowledge Engineering.

→ SE's or AIE's will take a problem & try to figure out how to distill it down into knowledge that is representable by a computer.

Clue: Find the murdrer, murder room & weapon as well.

People	Rooms	Weapon
Col. Mustard	Ballroom	Knife
Prof. Plum	Kitchen	Revolver
Ms. Scarlet	Library	Wrench



Propositional Symbols : It ~~could~~ be Some Symbol that can either be true or false.

→ In this case, it correspond with each possible thing that can be inside envelope.

Knowledge : (mustard v plum v scarlet)
(ballroon v library v kitchen)
(Knife v revolver v Wrench)

¬plum
¬mustard v ¬library v ¬revolver

→ **Inference Rules:** Some sort of rules that we can apply to take Knowledge that already exists & translate it into new forms of Knowledge.

- **Modus Pones** → If it is raining, then Harry is inside $(\alpha \to \beta)$

  It is raining. (α)

  $\overline{}$

  Harry is inside. (β)

- **And Elimination** → $\dfrac{\text{Harry is friend with Ron \& Hermione.}}{\text{Harry is friend with Hermione.}}$ $(\alpha \wedge \beta)$ / (α)

- **Double Negation Elimination** → $\dfrac{\text{It is not true that Harry did not pass the test.}}{\text{Harry passed the test}}$

  $$\frac{\neg(\neg\alpha)}{\alpha}$$

- **Implication Elimination** → $\dfrac{\text{If it is raining, then Harry is inside.}}{\text{It is not raining or Harry is inside.}}$ $(\alpha \to \beta)$ / $\neg\alpha \vee \beta$

- **Biconditional Elimination** →
  $$\frac{\alpha \leftrightarrow \beta}{(\alpha \to \beta) \cancel{\wedge} \vee (\beta \to \alpha)}$$

- **De Morgan's Law** →
  $$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta}$$

- **Distributive property** →
  $$\frac{(\alpha \wedge (\beta \vee \gamma))}{(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)}$$

Theorem Proving →

- initial state : starting Knowledge base
- actions : inference rules
- transition model : new Knowledge base after inference.
- goal test : check statement we are trying to prove.
- path cost function : number of steps in proof.

Clause : a disjunction of literals
e.g. $P \vee Q \vee R$

Conjuctive Normal Form (CNF) → logical sentence that is a conjuction of clauses

e.g. $(A \vee B \vee C) \wedge (D \vee \neg E) \wedge (F \vee G)$

Conversion to CNF →

- Eliminate biconditionals
  - turn $(\alpha \leftrightarrow \beta)$ into $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
- Eliminate implications
  - turn $(\alpha \rightarrow \beta)$ into $\neg \alpha \vee \beta$
- Move $\neg$ inwards using De Morgan's Laws
  - e.g. turn $\neg(\alpha \wedge \beta)$ into $\neg \alpha \vee \neg \beta$
- Use distributive law to distribute $\vee$ wherever possible.

$(P \vee Q) \rightarrow R$

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

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

$= (\neg P \vee R) \wedge (\neg Q \vee R)$

Python
#Search OO

Code,
(BST).

Date ______
Page ______

Inference by Resolution →

→ • To determine if $KB \models \alpha$:
 • Check if $(KB \wedge \neg\alpha)$ is a contradiction?
 • if so, then $KB \models \alpha$.
 • otherwise, no entailment.

→ • To determine if $KB \models \alpha$:
• Convert $(KB \wedge \neg\alpha)$ to CNF.
• Keep checking to see if we can use resolution to produce a new clause.
• If ever we produce the empty clause (equivalent to False), we hav a contradiction, & $KB \models \alpha$.
• otherwise, if we can't add new clauses, no entailment.

Teacher's Signature ....

• Introduction to predicate logic

→ It is an extension of proposition logic

• Quantifer: It is word phrase refers to quantity of element.

→ It is of two types-

Universal Quantifier ($\forall$) for all, every

Existential Quantifier ($\exists$) some, any

x is smaller than 10.

$P(x)$ → x is smaller than.

$Q(x,y)$ → x is smaller than y.

$\boxed{\forall x\, P(x)}$ = True, if every domain value satisfies condition.

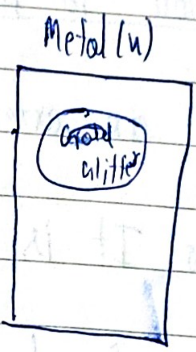
for all value of

$\exists$ is $Q(x)$ = True, if even one value satisfy condition.

Domain: $\{1, 3, 5, 7\}$ for $\forall$ T T T T , Domain: $\{1, 3, 5, 7, 15\}$ for $\exists$ T T T T F

- <u>First order predicate logic</u>

Q: Not all that Glitter is Gold.



Glitter (n) -> n is glitter
Gold (n) -> n is gold
$\forall$ (n) -> for all n

$\boxed{\forall n \text{ Glitter}(n) \rightarrow \text{Gold}(n)}$ or,

$\boxed{\forall n \underbrace{\text{Glitter}(n)}_{A} \wedge \underbrace{\text{Gold}(n)}_{B}}$

A	B	O/P
0	0	T
0	1	T
1	0	F
1	1	T

A	B	O/P
0	0	0
0	1	0
1	0	0
1	1	1

for all or every, use implication.
for some or any, use OR.

Sol: $\Rightarrow \forall n \text{ Glitter}(n) \rightarrow \text{Gold}(n)$

$\Rightarrow \sim\{\forall n (\sim \text{Glitter}(n) \cup \text{Gold}(n))\}$

$= \exists n \text{ Glitter}(n) \wedge \sim \text{Gold}(n)$

$\sim\{\sim p\} = p$

$\sim\{p\} = \sim p$

$\sim\{\sim p \vee q\} = p \wedge \sim q$

$\sim\{p \wedge \sim q\} = \sim p \vee q$

$A \rightarrow B = A' + B$ or $A' \vee B$

$\sim\forall = \exists$

$\sim\exists = \forall$

$\sim(\forall n\ (P(n) \wedge Q(n)))$

$\Rightarrow$ ~~∃n(~~

$\exists n\ (\sim P(n) \vee \sim Q(n))$

- <u>Semantic Network frames</u> → A meaningful graph Consist of relationship b/w object.
  These represent Knowledge in the form of graphical networks.

E.g. Tom is a cat

Mammal

Tom ...