

Lab-7

→ Propositional Logic :-

$P \rightarrow Q$ (If P is true, Q then Q is true)

(We know P is true)

Knowledge Based :-

1) Alice is mother of Bob.

2) Bob is the father of Charlie.

3) ♂ father is a parent.

4) A mother is a parent.

5) All the parents have children

6) If someone is a parent, their children are siblings.

7) Alice is married to David.

Hypothesis :-

"Charlie is a sibling of Bob."

→ Propositional logic :-

1) $M(A, B)$: Alice is mother of Bob

2) $F(B, C)$: Bob is father of Charlie

3) $\text{Parent}(x)$: x is a parent

4) ~~$\text{Child}(y, x)$~~ : y is a child of x.

5) $\text{Siblings}(x, y)$: x & y are siblings

6) $\text{Married}(A, D)$: Alice is married to David

7) $\text{Parent}(x)$ has children ~~(y)~~ who are siblings (x, y)

Logical reasoning

8-dec

1) From statement 1 & 4

~~A(a)~~ $M(A, B) \in (y, x) \rightarrow$ Alice is parent

2) From 2 & 4 :

$P(B, C) \in (y, x) \rightarrow$ Bob is a ~~parent~~ parent

3) From 1 & 2 & 7 :

~~$M(A, B) \wedge P(B, C) \wedge (x \in (x, y)) \rightarrow$~~

Bob & Charles are
siblings.

From
1/1/2021

\rightarrow Charles is father

$((P \rightarrow Q) \leftarrow (D \wedge S \wedge (X \rightarrow D)) \vee \neg S \vee \neg V$

(and) notes

(from) notes

$((Q \wedge X) \rightarrow (A \wedge (B \wedge (C \wedge D))) \leftarrow (X \rightarrow A) \wedge V$

(logically sound) notes

(intuitively sound, note 62) the others

Lab - 7

Code :-

$((\text{push } x) \text{ insert}) \times 5$

Class knowledge :

def - note (self): $\text{print } (\text{1}) \text{ bureaucrat work}$ ①

Self - rules = [] \vee print (push, x) insert

Self - facts = set().

def add - fact (self - fact):

self - facts . add (fact) $\text{tell user about } (\text{C}) \text{ bureaucrat will}$ ②

def add - rule (self, premise, conclusion): $\text{tell user about } (\text{C})$

Self - rules . append ((premise, conclusion)),

def infer (self):

new - inference = True

while new - inference :

new - inference = False

For premise, conclusion in self . rules;

if all (fact in self . facts for fact in premise),

if conclusion not in self . facts:

~~self . facts . add (Conclusion)~~

new - inference = True

def entails (self, hypothesis):

~~return hypothesis in self . facts.~~

Kb = knowledge base

Kb.add_fact ("Alice is mother of Bob")

Kb.add_fact ("Bob is father of Alice")

Kb.add_fact ("Alice is mother of Bob")

Kb.add_fact ("Bob is father of Alice")

{ "((A,B) is child) & (A is female) & (B is male) & "Alice is mother of Bob" }

Kb.add_fact ("Bob is father of Alice")

{ "((B,A) is child) & (B is female) & (A is male) & "Bob is father of Alice" }

Kb.add_rule (["Bob is father of Charlie", "A father is a parent"], "Bob is a parent")

Kb.add_rule (["Alice is mother of Bob", "A mother is a parent"], "Alice is a parent")

["parent"] \rightarrow ["parent"] \wedge ["parent"]

Kb.add_rule (["Bob is a parent", "All parents have children"], "Charlie and Bob are siblings")

Kb.infer ()

hypothesis = "Charlie and Bob are siblings"

if Kb.entails (hypothesis):

print ("{} hypothesis" is entailed by knowledge base")

else:

Demonstrate ("{} hypothesis" is not entailed by kb")