Assignment

Shubham Shankar
1001761068

Task 1

Part a)

	KB	S_1	
①	T	T	✓
②	F	T	
③	T	T	✓
④	F	T	
⑤	F	F	
⑥	F	F	
⑦	T	T	✓
⑧	F	F	

$KB \models S_1$, KB entails S_1, when ever KB is true & S_1 is also true.

In the above ~~diag~~ Table, for ①, ③, ⑦ , S_1 is true when KB is True.

Thus we can say that $\boxed{\text{KB entails } S_1}$

Part b) Not (KB) will entail Not (S_1) if & only if Not(S_1) is true, for when Not (KB) is true

	Not (KB)	Not(S1)	
1)	F	F	
2)	T	F	X
3)	F	F	
4)	T	F	X
5)	T	T	
6)	T	T	
7)	F	F	
8)	T	T	

From the above table, we see that there are 5 places where Not(KB) is true, But Not(S_1) is not true in all those 5 places.

This means that Not(Kb) does not entail Not(S_1)

∴ Summary,

KB entails S_1
Not (KB) doesnot entail S_1.

2

Task 2

There are 2 cases where KB is false

Case 1) : A = true, B = false, C = false, D = true
Case 2) : A = false, B = false, C = true, D = false.
in all other cases KB is true.

For the given KB, Let X be the CNF

$$X = ((A \cap \neg B \cap \neg C \cap D) \cup (\neg A \cap \neg B \cap C \cap \neg D))'$$

$$X = ((A \cap \neg B \cap \neg C \cap D)' \cap (\neg A \cap \neg B \cap C \cap \neg D)')$$

$$X = ((\neg A \cup B \cup C \cup \neg D) \cap (A \cup B \cup \neg C \cup D))$$

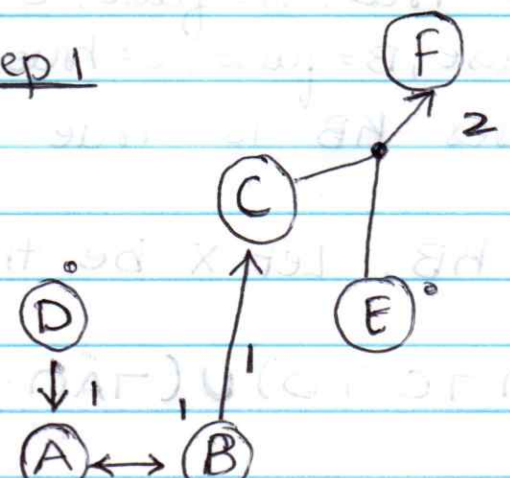
3

Task 3

A <=> B
B => C
D => A
C And E => F
E
D

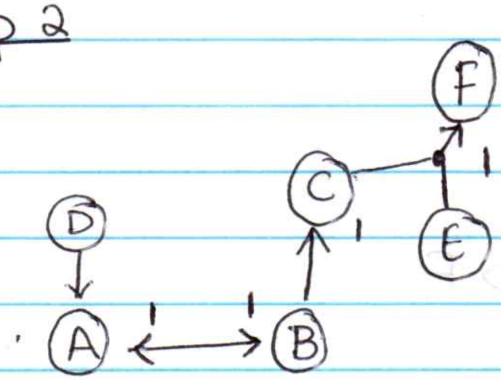
i) <u>Forward chaining</u>

Step 1



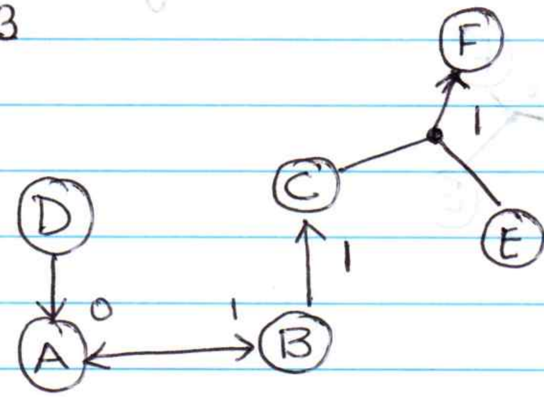
Initially Arc D & E have Value zero. Then in next Step Reduce the Arc value which has E as a starting point.

Step 2



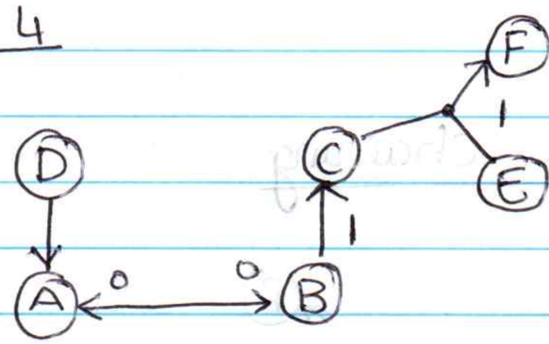
Now here we see D has Value zero. So Reduce the Value of Arch which has D as starting point.

Step 3



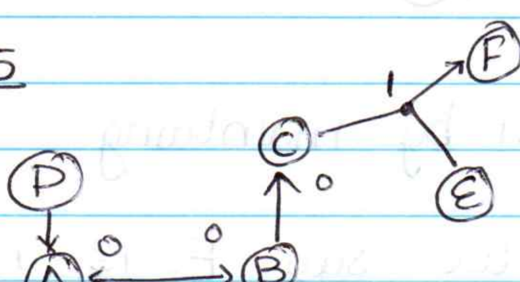
here A has Value zero, reduce the Value of Arc which has A as Starting point.

Step 4

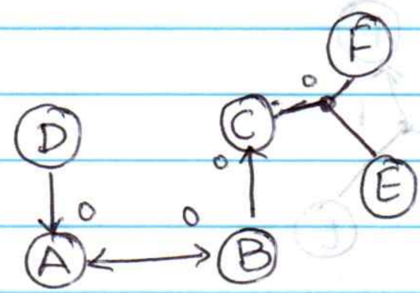


here B has Value zero, reduce the Value of Arc with B as starting point

Step 5



here C has value zero, now reduce the Arc, with C as a starting point

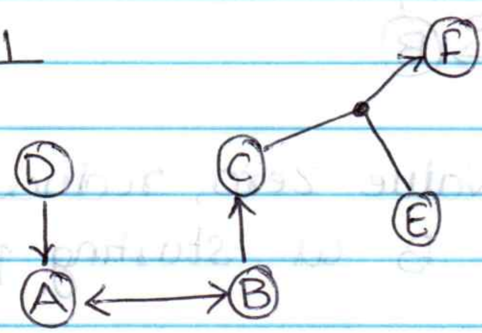


Finally F becomes zero,

Thuse we can say that $\boxed{\text{KB entails F}}$

ii) <u>Backward Chaining</u>

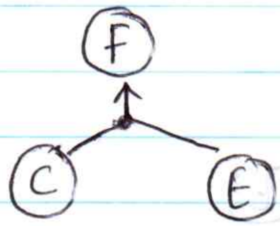
<u>Step 1</u>



we start by maintaing goal stack.

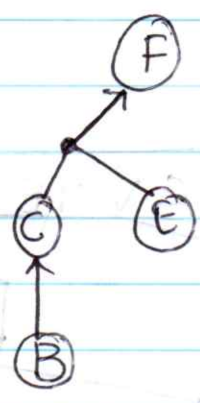
Initially we say F is true

So next we will see for the Rule that result in F being True

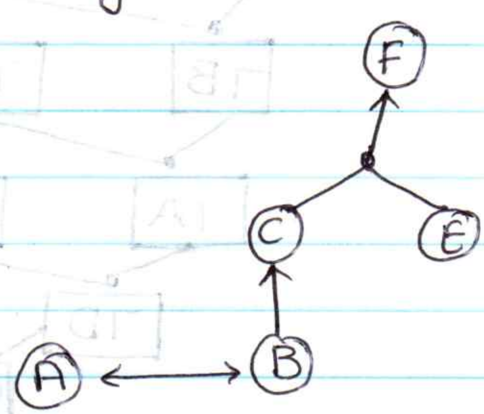


Now we add C & E to the stack
E is already True
We will Look for result that make C true

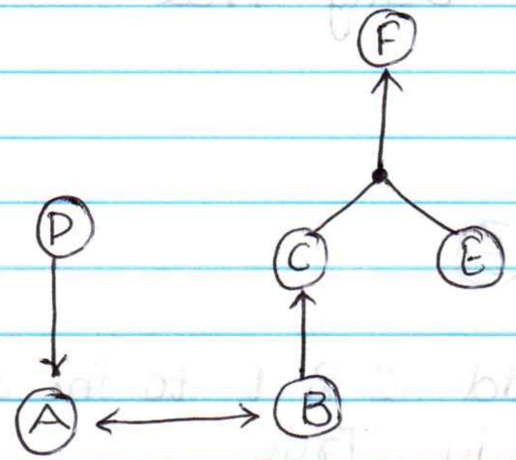
So now we add B to stack



similarly A is added to stack

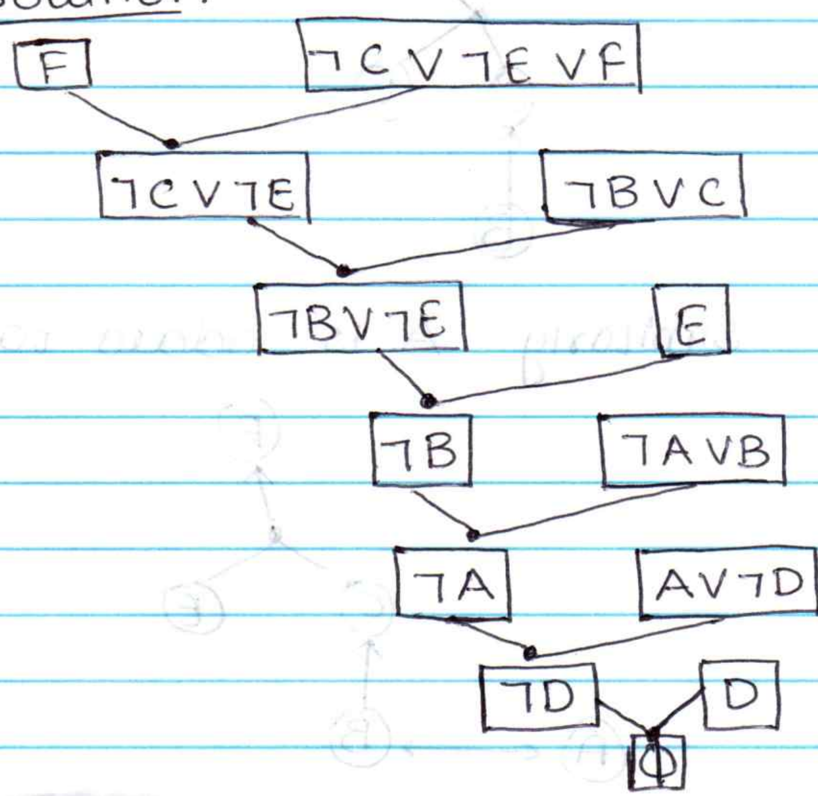


Now we add D to the stack.



Thus we can say $\boxed{\text{"KB entails F"}}$

iii) Resolution



(4) Task 4

(a) Constants : John, Mary, May1 2017, May 2 2017
$10000, May 3 2017

Predicates :- rain(X) - rains on X
give(w, x, y, z) = w give y to x on z
mow(x, y) = x mow ~~laan~~ on y

Contract
rain(May 1 2017) ⇒ give(John, mary, Check of $10000, May 2 2017)

give(John, mary, Check of $10000, May 2 2017)
⇒ Mow(marry, May 3 2017)

(b) What truly happend :

¬rain(May 1, 2017)
give (John, Mary, Check of $10000, May 2 2017)
mow (marry, may 3 2017)

c) No, the contract was not violated as per FOL because first Event is always true & Event give & mow took place.

(d) Taking Constant & predicate from part (a)

Contract

rain(may 1 2017) = A

give(John, mary, $10000, May 2 2017) = B

mow(marry, May 3 2017) = C

Conversion

$A \Rightarrow B$

$B \Rightarrow C$

Convertion for what actually happend

$\neg A$

B

C

Task 5

(5) **Predicates**

Adult(x) : x is adult
Child (x) : x is child
Boat (x) : x is boat
Onleyt(x) : x is on leyt side.
Onright(x) : x is on right side.

Initial state

Boat(b) $\cap$ Onleyt(b) $\cap$ Child (c_1) $\cap$ onleyt (c_1)
Child (c_2) $\cap$ onleyt (c_2) $\cap$ Child (c_3) $\cap$ Onleyt(c_3)
$\cap$ adult(a_1) $\cap$ onleyt (a_2) $\cap$ adult (a_2) $\cap$ Onleyt(a_2)
$\cap$ Adult (a_3) $\cap$ onleyt (a_3)

Goal state

Onright (b) $\cap$ child (c_1) $\cap$ onright (c_1) $\cap$ child (c_2)
$\cap$ onright(c_2) $\cap$ child(c_3) $\cap$ onright (c_3) $\cap$ adult(a_1)
$\cap$ onright(a_2) $\cap$ adult (a_2) $\cap$ onright (a_2)
$\cap$ adult (a_3) $\cap$ onright (a_3).

Operation

Action : moveright two (x, y, b)
precondition: Child(x) $\cap$ child(y) $\cap$ onleft (x)
$\cap$ Onleft(y) $\cap$ onleft(b) $\cap$ boat(b)
Effects: Onright (x) $\cap$ onright (y) $\cap$ onright(b)
$\cap$ not (Onleft (x)) $\cap$ not (onleft (y)) $\cap$ not(onleft(b))

Action: moveright (x, b)
precondition: Onleft (x) $\cap$ onleft (b) $\cap$ boat (b)
Effects: Onright(x) $\cap$ onright (b) $\cap$ not(onleft(x))
$\cap$ not (onleft (b))

Action : move right two (x y b)
precondition: child (x) $\cap$ Adult (y) $\cap$ onleft (x)
$\cap$ Onleft (y) $\cap$ onleft (b) $\cap$ Boat (b)
Effects : onright (x) $\cap$ onright (y) $\cap$ onright (b)
$\cap$ not (onleft (x)) $\cap$ not (onleft (y)) $\cap$ (not (onleft(b))

Action: move left two (x y b)
precondition: child (x) $\cap$ child(y) $\cap$ onright(x) $\cap$ onright(y)
$\cap$ (onright (b) $\cap$ Boat (b)
Effects : onleft (x) $\cap$ onleft(y) $\cap$ onleft (b) $\cap$ not(onright (x))
$\cap$ (not(onright (y)) $\cap$ not (onright (b))

Action: move lyt (x b)
precondition: Onright (x) ∩ Onright(b) ∩ Boat(b)
Effect: Onlyt (x) ∩ onlyt(b) ∩ not(onright (x)) ∩ not (onright(b))

Action: move lyt two (x y b)
precondition: Child (x) ∩ adult (y) ∩ onright (x) ∩ onright (y)
∩ Onright (b) ∩ boat (b)
Effect: onlyt (x) ∩ onlyt (y) ∩ onlyt (b) ∩ not (on right (x))
∩ (not (onright (y)) ∩ not(onright (b))

Complete Plan

move lyt two (c_1, c_2, b)
moveright (c_1, b)
move lyt (c_1, b)
move right two (c_1, a_1, b)
move lyt (c_1, b)
move right two (c_1, a_2 b)
move lyt (c_1, b)
move right two (c_1, a_3 b)
move lyt (c, b)
move right (c_1 c_3 b)

(6) Task 6

In JUNGLE World there are
4 predicates
4 arguments
5 constants.

4 predicates take [1 4] arguments.
Number of ways to assign 5 constant

$$= [4 \times 5^1 \quad 4 \times 5^4]$$
$$[20 \quad 2500]$$

The PDDL state is defined by listing all the predicates that are true

For n predicates, the possible states are

$$nc_0 + nc_1 + nc_2 \cdots nc_n = \sum_{i=0}^{n} nc_i = 2^n$$

$\therefore$ tight bound on the number of State in JUNGLE World is

$$\left[\sum_{i=0}^{20} 20c_i \quad \sum_{i=0}^{2500} 2500c_i \right]$$

$$= [2^{20} \quad 2^{2500}]$$

7) Task 7

- Execution Monitoring | Online Replanning :-
For this there is no need to make any modification as it would replan the entire Scenario.
If the goal of an action is not satisfied then the system replans the flow again from the current state

- Conditional planning :-

For this we need to modify

Action: move right (x, b)
Pre Condition: Onleft(x) ∩ Onleft(b) ∩ Boat(b)
Effect : (onright(x) ∩ onright(b) ∩ not (onleft(x) ∩ (not (onleft(b))) ∨ (onleft(x) ∩ onleft(b))

Action: move left (x b)
pre condition: onright(x) ∩ onright(b) ∩ boat(b)
Effect: (onleft(x) ∩ onleft(b) ∩ not (onright(x) ∩ not (onright(b))) ∨ (onright(x) ∩ onright(b)))