Prem A. Jethwa

1001861810

- Assignment 3 -

* Task 1:

Part a:

KB does entail S1, because as per the entailment rule, if KB is true for any state, S1 is true too and if KB is false for any state, S1 is either true or false.

$$\therefore KB \models S1$$

Part b:

NOT (KB) does not entail NOT (S1). Because as per entailment conditions and also the given truth table; it can be concluded that if the truth value are negated, there are some states where the truth value of KB is true, S1 would be false which doesnot satisfy entailment. Hence NOT (KB) $\not\models$ NOT (S1)

NOT (KB)	NOT (S1)
False	False
True	False
False	False
True	False
True	True
True	True
False	False
True	True

* Task 2:

Cases when Knowledge base is FALSE.

Case1: When A is true, B is false, C is true, D is false

Case2: When A is false, B is false, C is true, D is true

Build a statement which is only false in the given models and true otherwise.

$\neg[(A \wedge \neg B \wedge C \wedge \neg D) \vee (\neg A \wedge \neg B \wedge C \wedge D)]$

Corollary to CNF.

- Apply DeMorgans;

$\neg(A \wedge \neg B \wedge C \wedge \neg D) \wedge \neg(\neg A \wedge \neg B \wedge C \wedge D)$

$(\neg A \vee \neg(\neg B) \vee \neg C \vee \neg(\neg D)) \wedge (\neg(\neg A) \vee \neg(\neg B) \vee \neg C \wedge \neg D)$

- Remove Double Negation;

$(\neg A \vee B \vee \neg C \vee D) \wedge (A \vee B \vee \neg C \vee \neg D)$

which is in CNF

A	B	C	D	R B
T	T	T	T	T
T	T	T	F	T
T	T	F	T	T
T	T	F	F	T
T	F	T	T	T
T	F	T	F	F
T	F	F	T	T
T	F	F	F	T
F	T	T	T	T
F	T	T	F	T
F	T	F	T	T
F	T	F	F	T
F	F	T	T	T
F	F	T	F	F
F	F	F	T	T
F	F	F	F	T

CNF (RB) : $\neg$(Row 6) $\wedge$ $\neg$(Row 13)

* Task 3 -

a. Forward Chaining:

For forward chaining the KB and α both need to be in HORN form.

KB → Converting KB to HORN form

$A \rightarrow C$ → $A \rightarrow C$

$B \Leftrightarrow C$ → $B \rightarrow C$ $\quad C \rightarrow B$

$D \rightarrow A$ → $D \rightarrow A$

$(B \wedge E) \rightarrow G$ → $(B \wedge E) \rightarrow G$

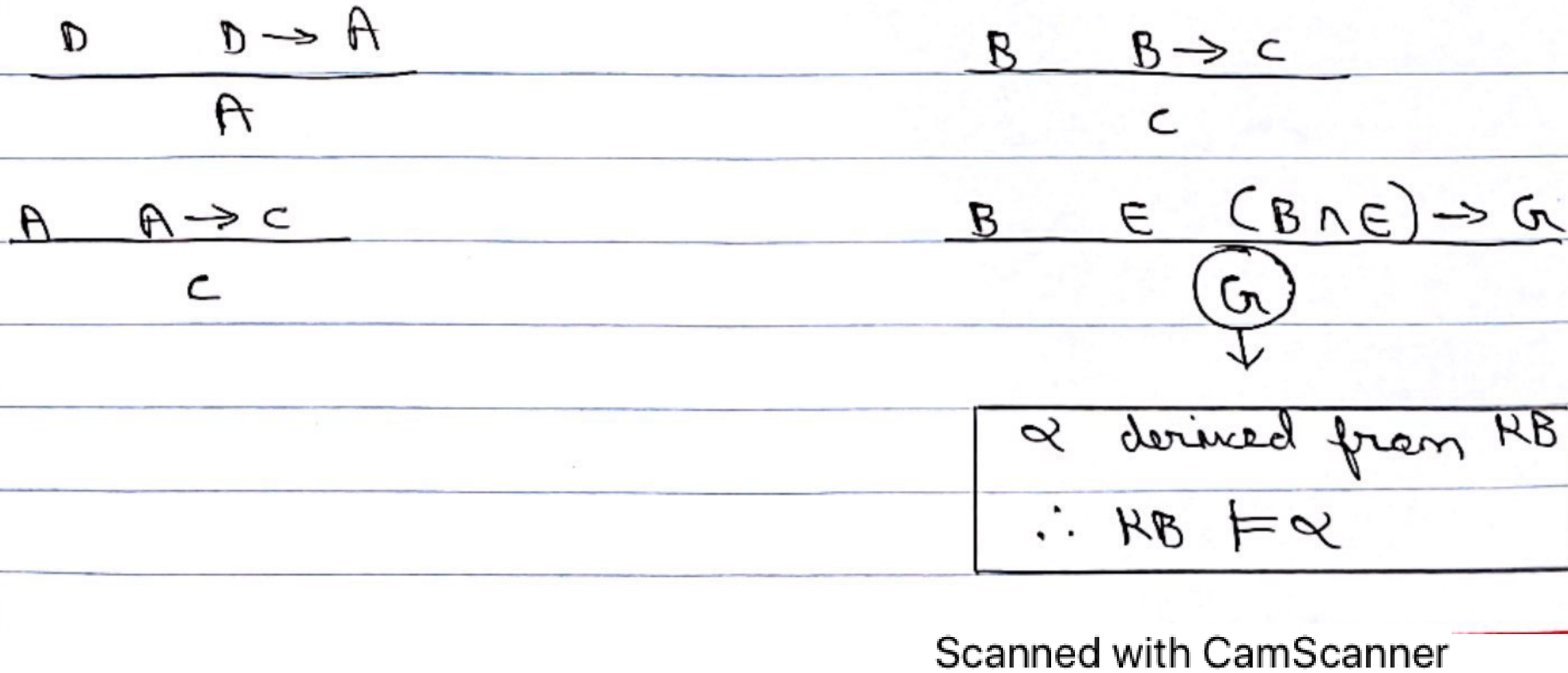
$B \rightarrow F$ → $B \rightarrow F$

E → E

D → D

α // G → α // G

Forward Chaining Steps:



b. Backward Chaining:

For backward chaining as well, the KB and α both need to be in HORN form.

KB

$A \rightarrow C$

$B \leftrightarrow C$

$D \rightarrow A$

$(B \wedge E) \rightarrow G$

$B \rightarrow F$

E

D

α // G

→ Converting KB to HORN form.

$A \rightarrow C$

$B \rightarrow C$ $\quad C \rightarrow B$

$D \rightarrow A$

$(B \wedge E) \rightarrow G$

$B \rightarrow F$

E

D

α // G

For backward chaining we need to find the dependent literals and add them to the stack.

It is done as follows:

A
C
B
G

→ D & E already there
∴ not added in stack.

Steps:

(I) $\dfrac{D \quad D \rightarrow A}{A}$

C
B
G

(II) $\dfrac{A \quad A \rightarrow C}{C}$

B
G

(III) $\dfrac{C \quad C \rightarrow B}{B}$

G

(IV) $\dfrac{B \quad E \quad (B \wedge E) \rightarrow G}{}$

α derived from KB $\therefore$ KB $\models \alpha$

c. Resolution:

The KB must be in CNF form for applying resolution.

$(A \rightarrow C) \wedge (B \Leftrightarrow C) \wedge (D \rightarrow A) \wedge ((B \wedge E) \rightarrow G) \wedge (B \rightarrow F) \wedge E \wedge D$

↳ given

$(\neg A \vee C) \wedge (B \rightarrow C) \wedge (C \rightarrow B) \wedge (\neg D \vee A) \wedge [\neg (B \wedge E) \vee G] \wedge$

$(\neg B \vee F) \wedge E \wedge D$ ..... (eliminating $\rightarrow$ & $\Leftrightarrow$).

$(\neg A \vee C) \wedge (\neg B \vee C) \wedge (\neg C \vee B) \wedge (\neg D \vee A) \wedge [(\neg B \vee \neg E) \vee G] \wedge$
$(\neg B \vee F) \wedge E \wedge D$ ..... (eliminating $\rightarrow$ 9 using D'Morgan)

$(\neg A \vee C) \wedge (\neg B \vee C) \wedge (\neg C \vee B) \wedge (\neg D \vee A) \wedge [\neg B \vee \neg E \vee G] \wedge$
$(\neg B \vee F) \wedge E \wedge D$ ..... (CNF form).

Applying Resolution:

$(\neg A \vee C)$ $(\neg B \vee C)$ $(\neg C \vee B)$ $(\neg D \vee A)$ $(\neg B \vee \neg E \vee G)$ $(\neg B \vee F)$
E D $(\neg G)$
(compliment of α added)

$\frac{(\neg A \vee C)(\neg C \vee B)}{(\neg A \vee B)}$ $\frac{(\neg A \vee C)(\neg D \vee A)}{(C \vee \neg D)}$ $\frac{(\neg B \vee C)(\neg C \vee B)}{(\neg B \vee B)(C \vee \neg C)}$ $\frac{(\neg C \vee B)(\neg B \vee \neg E \vee G)}{(\neg C \vee \neg E \vee G)}$

$\frac{(\neg C \vee B)(\neg B \vee F)}{(\neg C \vee F)}$ $\frac{(\neg C \vee B)(C \vee \neg D)}{(B \vee \neg D)}$ $\frac{(\neg D \vee A)\ D}{A}$ $\frac{A\ (\neg A \vee C)}{C}$

$\frac{C\ (\neg C \vee B)}{B}$ $\frac{B\ (\neg B \vee \neg E \vee G)}{(\neg E \vee G)}$ $\frac{E\ (\neg E \vee G)}{G}$ $\frac{G\ (\neg G)}{\square}$

$\downarrow$

empty clause

$\therefore$ We can say using Resolution for proof by contradiction $KB \vee \neg\alpha \rightarrow$ is unsatisfiable

ie. $KB \models \alpha$ (KB entails α)

* Task 4:

Part a -

J → John ; M → Mary ; M → Monday [Constants]

Relations:

Rain (x) → It rains on x,

Check (x, y) → x gives y $100 check on Tuesday

Mows (x) → x mows lawn on Wednesday.

Contrast:

If it rains on Monday, then John must give Mary a check for $100 on Tuesday.

$$Rains(Mon) \rightarrow Check(J, M)$$

If John gives Mary a check for $100 on Tuesday, Mary must mow the lawn on Wednesday.

$$Check(J, M) \rightarrow Mows(M).$$

Part b:

Events:

It did not rain on Monday → ¬ Rains (Mon).
John gave Mary a check for $100 on Tuesday
→ Check (J, M)
Mary mowed the lawn on Wednesday → Mows (M)

Logical Statement :
(¬ Rains (Mon)) ∧ Check (J, M) ∧ Mows (M)

Part c:

All possible combinations of relations and constants defined above:

A_1 → Rains (J) ; A_2 → Rains (M) ; A_3 → Rains (Mon).
B_1 → Mows (J) ; B_2 → Mows (M) ; B_3 → Mows (Mon).
C_1 → Check (J, J) ; C_2 → Check (J, M), C_3 → Check (J, Mon)
C_4 → Check (M, J) ; C_5 → Check (M, M) ; C_6 → Check (M, Mon).
C_7 → Check (Mon, J) ; C_8 → Check (Mon, M) ; C_9 → Check (Mon, Mon)

Contract:

If it rains on Monday, then John must give Mary a check for $100 on Tuesday.

$$A_3 \rightarrow C_2$$

If John gives Mary a check for $100 on Tuesday, Mary must mow the lawn on wednesday.

$C_2 \rightarrow B_2$

Events:

It did not rain on Monday $\rightarrow \neg A_3$

John gave Mary a check for $100 on Tuesday $\rightarrow C_2$.

Mary mowed the lawn on Wednesday $\rightarrow B_2$.

Part d:

Here we need to check if events entail contract. The events and contract can be written as follows:

Events: $(\neg A_3) \wedge C_2 \wedge B_2$

Contract: $(A_3 \rightarrow C_2) \wedge (C_2 \rightarrow B_2)$

Checking if events entail contract using Truth Table.

A_3	B_2	C_2	$(\neg A_3) \wedge C_2 \wedge B_2$	$(A_3 \rightarrow C_2) \wedge (C_2 \rightarrow B_2)$
T	T	T	F	T
T	T	F	F	F
T	F	T	F	F
T	F	F	F	F
F	T	T	T	T
F	T	F	F	F
F	F	T	F	T
F	F	F	F	T

Here we can see that for all values where Events is True, the contract is True as well.

$\therefore$ Events $\models$ Contract.

$\therefore$ Contract is not violated.

* Task 5 :

I: Taller (x, John) ; Taller (Bob, y)

∴ Unifier θ can be → {x/Bob , y/John}

II: Taller (y, Mother (x)) ; Taller (Bob, Mother (Bob))
Here y can be Bob. So that 1st part is same.
For 2nd part, Mother (x) = Mother (Bob).
∴ x = Bob or x = Sibling (Bob).

∴ Unifier θ can be → {x/Bob, y/~~John~~ Bob} or
{x/Sibling (Bob), y/Bob}

III: Taller (Sam, Mary) ; Shorter (x, Sam)
Assuming that shorter (x, Sam) means x is shorter than Sam & Taller (Sam, Mary) means Sam is taller than Mary.
Here 2nd predicate implies that sam is taller than x.
∴ We can say that Taller (Sam, x) have a unifier → which is true.
θ = {x/Mary}

∴ Taller (Sam, Mary) & Taller (Sam, x)
have a unifier → θ = {x/Mary}

IV. Shorter (x, Bob) ; Shorter (y, z).

An unifier exists for above two predicates

→ $\theta = \{x/y, z/Bob\}$

V. Shorter (Bob, John) ; Shorter (x, Mary)

Here even if we substitute x as Bob, we would still end up with two different predicates.

∴ We can say that no unifier exists for these 2 predicates.

* Task 6 -

Constants:

P_1, P_2, P_3, P_4, B, Left, Right

Predicates:

On (x, y) : x is on y bank

Is Boat (x) : x is boat

Is Child (x) : x is child

Initial State:

On (P_1, Left) ∧ On(P_2, Left) ∧ On (P_3, Left) ∧
On (P_4, Left) ∧ On (B, Left) ∧ Is Boat (B) ∧
Is Child (P_3) ∧ Is Child (P_4).

Goal state:

On (P_1, Right) ∧ On (P_2, Right) ∧ On (P_3, Right) ∧
On (P_4, Right).

Actions:

1. Move_one_left2right (x, y):

Move one person x from left to right using boat y.

PRE: On(x, left) ∧ On (y, left) ∧ Is Boat (y).

EFF: On(x, Right) ∧ On(y, Right) ∧ ¬On (x, left) ∧
¬ On(y, left)

2. Move_one_right2left (x, y):

Move one person x from right to left using boat y.

PRE : On (x, Right) ∧ On(y, Right) ∧ IsBoat (y)

EFF : On (x, left) ∧ On (y, Left) ∧ ¬ On(x, Right) ∧ ¬ On(y, Right).

3. Move_two_left2right (x, y, z):

Move adult/ child x & child y from left to right using boat z.

PRE: On (x, left) ∧ On (y, Left) ∧ On(z, left) ∧ IsBoat (z) ∧ Is Child (y).

EFF: On (x, Right) ∧ On (y, Right) ∧ On(z, Right) ∧ ¬ On(x, Left) ∧ ¬ On(y, Left) ∧ ¬ On(z, Left).

4. More_two_right2left (x, y, z):

Move adult/ child x & child y from right to left using boat z.

PRE: On (x, Right) ∧ On(y, Right) ∧ On (z, Right) ∧ IsBoat (z) ∧ Is child (y).

EFF: On(x, Left) ∧ On(y, Left) ∧ On(z, left) ∧ ¬ On(x, Right) ∧ ¬ On(y, Right) ∧ ¬ On(z, Right)

A possible Plan:

Plan:

Move-two-left 2 right (P_1, P_2, B)

Move-one-right 2 left (P_2, B)

Move-two-left 2 right (P_3, P_2, B)

Move-one-right 2 left (P_2, B)

Move-two-Left 2 right (P_4, P_2, B)

Status after taking action:

Left Side	Right Side
P_1, P_2, P_3, P_4, B	→ (Initial State)
P_3, P_4	P_1, P_2, B
P_2, P_3, P_4, B	P_1
P_4	P_1, P_2, P_3, B
P_2, P_4, B	P_1, P_3
	P_1, P_2, P_3, P_4, B (Goal)

* Task 7:

Execution Monitoring / Online replanning:

In this type of planning we do not need to modify the action definations. We can keep them as they were in the deterministic case.

Justification -

In this type of planning, instead of modifying the action definations we do a few additional things before performing any action to check if the state of the environment is same as it is expected to perform the action.

There are 3 ways to do so i.e. Action monitoring, Plan monitoring & Goal monitoring. In the given scenario, there is an unexpected environment change happening i.e. the boat can be blown, off course if there is one person in the boat which can be used as an advantage to come up with a better plan. So the best way to monitor the world would be Goal Monitoring.

Conditional Planning:

In this type of planning we need to consider the unexpected changes in the world in the action definitions. Following are the only required changes in the two action definition.

Move-one-left2Right (x,y):

Move one person x from left to right using boat y.

PRE: On(x, left) ∧ On(y, Left) ∧ IsBoat(y).

EFF: [On(x, Right) ∧ On(y, Right) ∧ ¬ On(x, Left) ∧ ¬ On(y, Left)] ∨ [On(x, Left) ∧ On(y, Left)]

Move-one-right2left (x,y):

Move one person x from right to left using boat y.

PRE: On(x, Right) ∧ On(y, Right) ∧ IsBoat(y).

EFF: [On(x, Left) ∧ On(y, Left) ∧ ¬ On(x, Right) ∧ ¬ On(y, Right)] ∨ [On(x, Right) ∧ On(y, Right)]

Here, these were the only two location/action definitions that required change because the effects of the other two

actions won't be affected considering the given information about the uncertain world changes. (ie." if there is only one person in the boat, the boat can be blown off course and end up back on the side it originally started from").

* Task 8 -

Given:

No. of predicates, $n = 3$; each predicate takes max 4 arguments.

i.e. $a \Rightarrow 1 \leq a \leq 4$

No. of constants, $m = 5$

From given information,
if no. of constants are 5 & no. of arguments a are $1 \leq a \leq 4$.
then no. of possible arguments for 1 predicate can be given as,

$$5^1 \leq \text{pass-args} \leq 5^4$$

As it is given that no. of predicates = 3.
∴ we can find no. of possible arguments for 3 predicates as follows,

$$(3 \times 5^1) \leq \text{pass-args-3} \leq (3 \times 5^4)$$
$$15 \leq \text{pass-args-3} \leq 1875$$

Now, to find possible no. of states, we know that a predicate can be either true or false ie. it can have 2 possible values.

$\therefore$ no. of states $= 2^{\text{pass\_args\_3}}$

And thus, no. of possible stages | states is,

$$2^{15} \leq \text{no. of states} \leq 2^{1875}$$

$$\text{i.e. } 32768 \leq \text{no. of states} \leq 2.69 \times 10^{564}$$