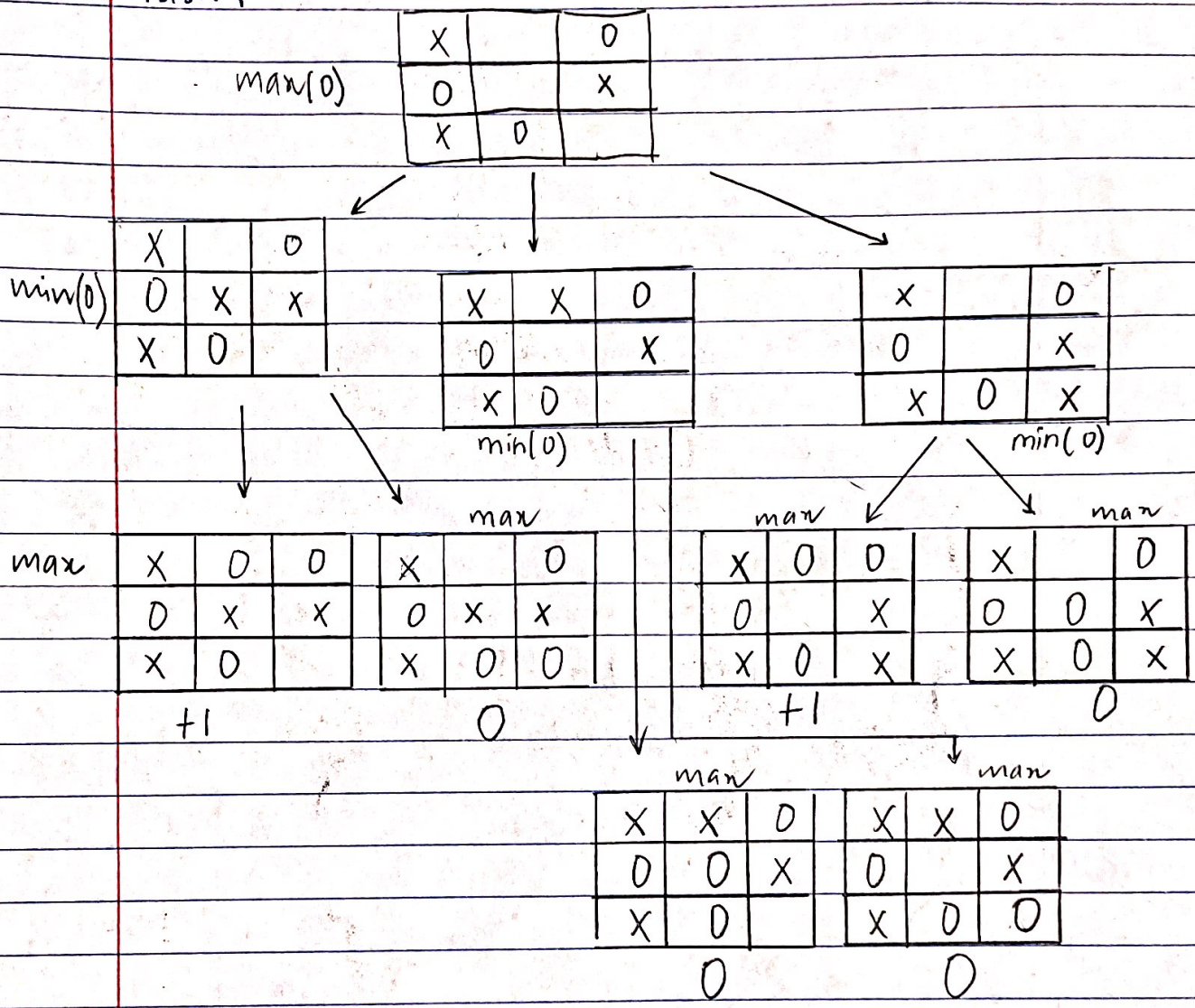
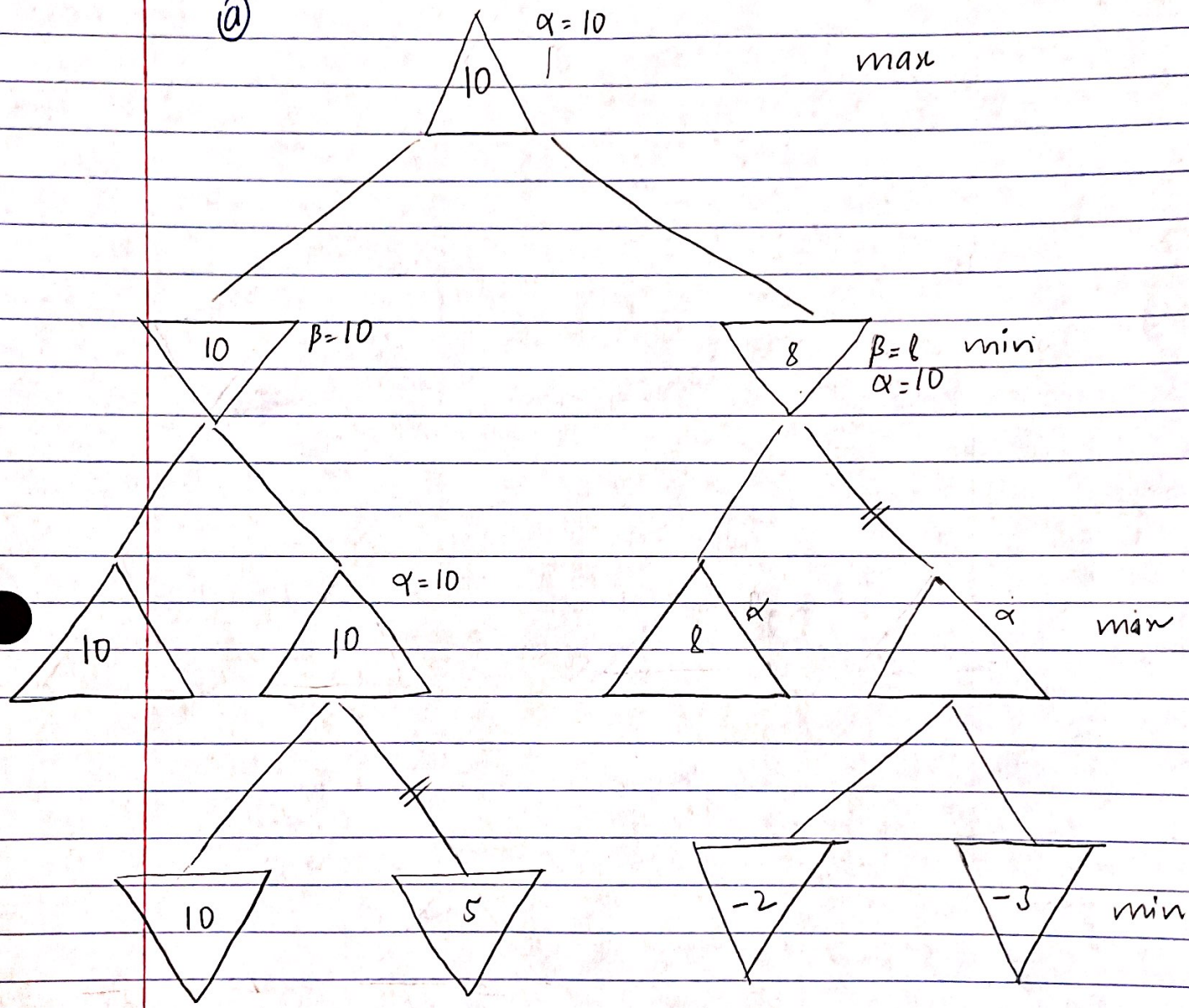
Assignment 3

Task 1



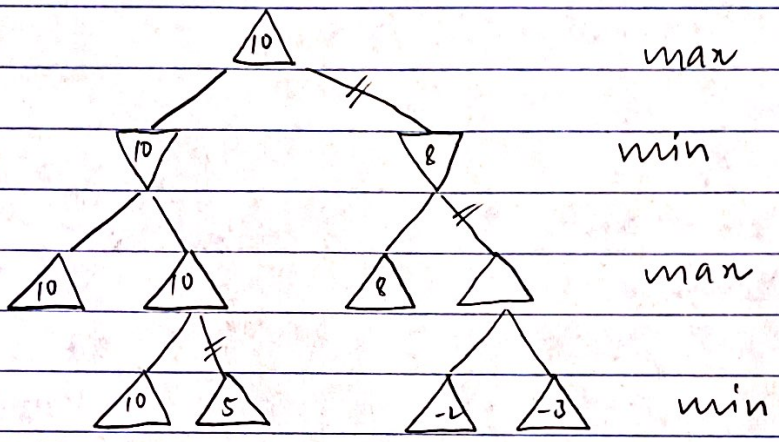
α ↑
β ↓
$\alpha \geq \beta$

Task 2
(a)



Task 2 (b)

Ans If we know that max utility value is 10, then any max node can stop exploring after it finds a Max value of 10



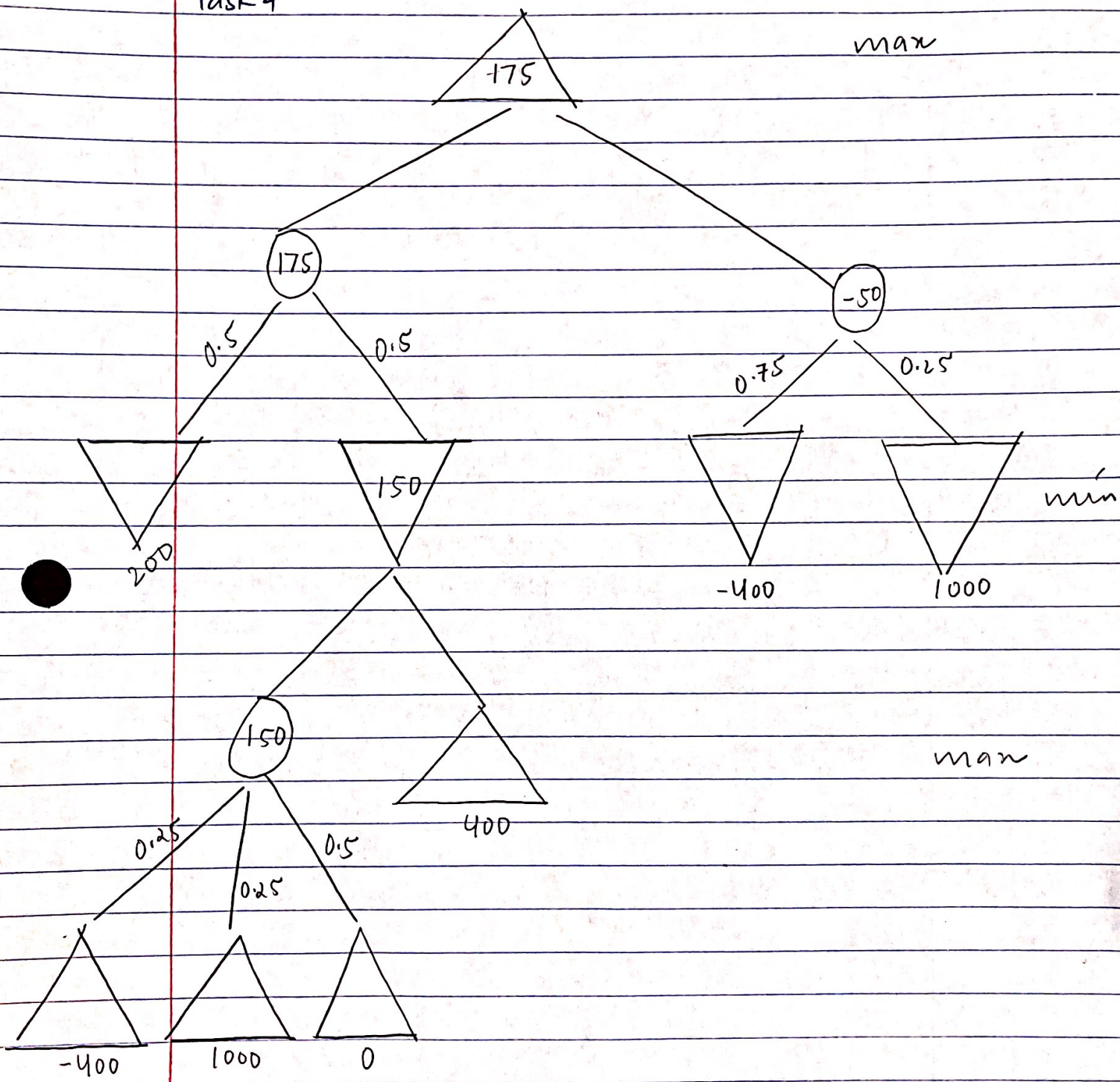
Here, maximum utility is 10, all the other value on the right side will be pruned.

Task 3

Ans=> Regular minmax will give the optimal decision. However, we can use DeepGreenMove(s) to reduce the amount of work we have to do.

function MinValue(state) returns a utility value
if TerminalTest(state) then return Utility(state)
else
return MaxValue(DeepGreenMove(s))

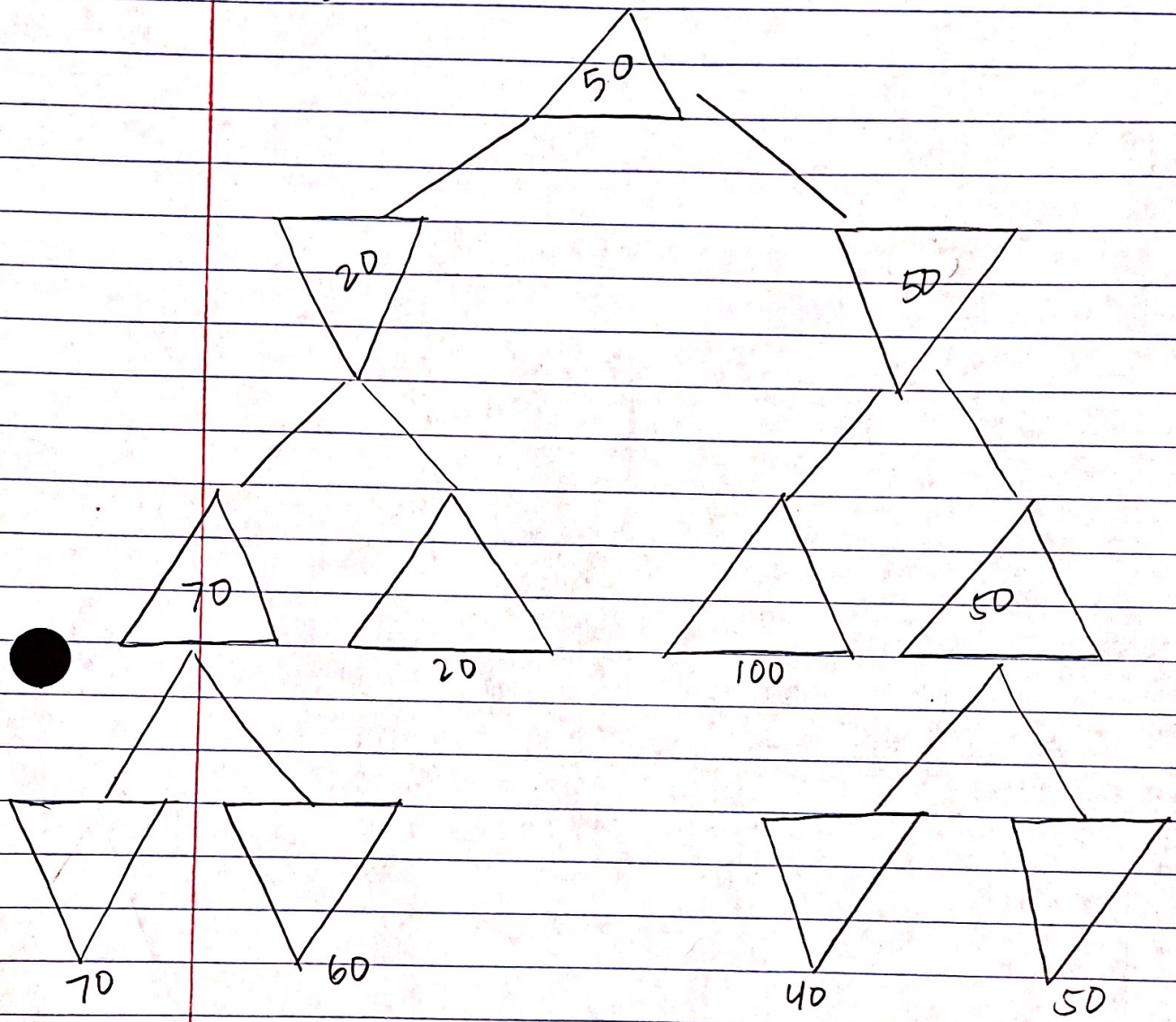
Task 4



lowest possible outcome = -50

highest possible outcome = 175

Task 5



The best possible outcome for MAX player is 100 if opponent doesnot try to minimize.

The worst possible outcome for MAX player is 50 if the opponent try to minimize

Task 6

Ans:

```
function CHECK-EQUIVALENCE(KB1, KB2)
returns true or false
{
    if TT-entails(KB1, KB2) and TT-entails(KB2, KB1)
        return true
    else
        return false
}
```

Task 7

a. Yes, KB entail S1 because for all the values where KB is true, S1 is also true (Rows: 1, 3, 7)

b. There are certain rows where NOT(KB) is true but NOT(S1) is not true. So, NOT(KB) doesnot entail statement NOT(S1). (Rows: 2, 4)

Task 8

Cases	A	B	C	D	KB
1	T	F	T	T	F
2	F	F	T	F	F

In other cases, where KB is true

Cases	A	B	C	D	KB
1st	F	T	F	F	T
2nd	T	T	F	T	T

Conjunctive Normal Form (CNF)

Applying De-Morgan's law

For

1st case: $\neg A \cup B \cup \neg C \cup \neg D$

2nd case: $A \cup B \cup \neg C \cup D$

Task 9

$A \Rightarrow B$

$B \Leftrightarrow C$

$D \Leftrightarrow A$

$E \Rightarrow D$

C AND $E \Rightarrow F$

E

1) Forward chaining

i) $\dfrac{E \qquad E \Rightarrow D}{D}$

ii) $\dfrac{D \qquad D \Rightarrow A}{A}$

iii) $\dfrac{A \qquad A \Rightarrow B}{B}$

iv) $\dfrac{B \qquad B \Leftrightarrow C}{C}$

v) $\dfrac{(C \wedge E) \Rightarrow F}{F}$

2) Backward chaining

i) $\dfrac{(C \wedge E) \Rightarrow F}{F}$

ii) $\dfrac{B \qquad B \Leftrightarrow C}{C}$

iii) $\dfrac{A \qquad A \Rightarrow B}{B}$

iv) $\dfrac{D \qquad D \Rightarrow A}{A}$

v) $\dfrac{E \qquad E \Rightarrow D}{D}$

By using
Modus Ponens

Task 9

3) Resolution

$(A \Rightarrow B) \wedge (B \Rightarrow C) \wedge (D \Rightarrow A) \wedge (E \Rightarrow D) \wedge (C \wedge E \Rightarrow F) \wedge E$

using implication elimination

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

Add $\neg F$

resolving

$\neg F \quad \neg C \vee \neg E \vee F$

$\rightarrow \neg C \vee \neg E$

$\neg C \vee \neg E \quad \neg B \vee C$

$\rightarrow \neg E \vee \neg B$

$\neg E \vee \neg B \quad \neg E \vee B$

$\rightarrow \neg E$

$\neg E \quad E$

$\square \rightarrow$ empty

$KB \neq F$ (false)

Task 10,

Constants : John, Mary, April, May, 10,000 check

Predicates :

rain(x) = it rains in x

gave(x, y, z) = x gave y a check of z

mow(x) = x mows the lawn

a. Contract :

rain(May) => gave(John, Mary, 10000)

gave(John, Mary, 10000) => mow(Mary)

b.

¬ rain(may)

gave(John, Mary, 10,000)

mow(Mary)

e.

- The contract is not violated because A => B is true always if A is false. It means if it rains then John must give the cheque to Mary. But, if it doesn't rain then its upto John whether he gives cheque to Mary or not.

Task 10

c) Symbols involved are

rain_May
gave_John_Mary_10000
mow_Mary

d) rain_May => gave_John_Mary_10000
gave_John_Mary_10000 => mow_Mary