Assignment 3 – Game Playing and Logic

Name : *Saranya Balasubramaniyan*
Net Id : *sxb9644*
Course and Section : *Artificial Intelligence 5360 – 900*

Task 1

Consider the tic-tac-toe board state shown in Figure 1. Draw the full minimax search tree starting from this state, and ending in terminal nodes. Show the utility value for each terminal and non-terminal node. Also show which move the Minimax algorithm decides to play for X. Utility values are +1if X wins, 0 for a tie, and -1 if O wins. (Note: X is the MAX player).

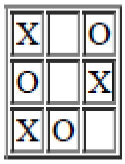
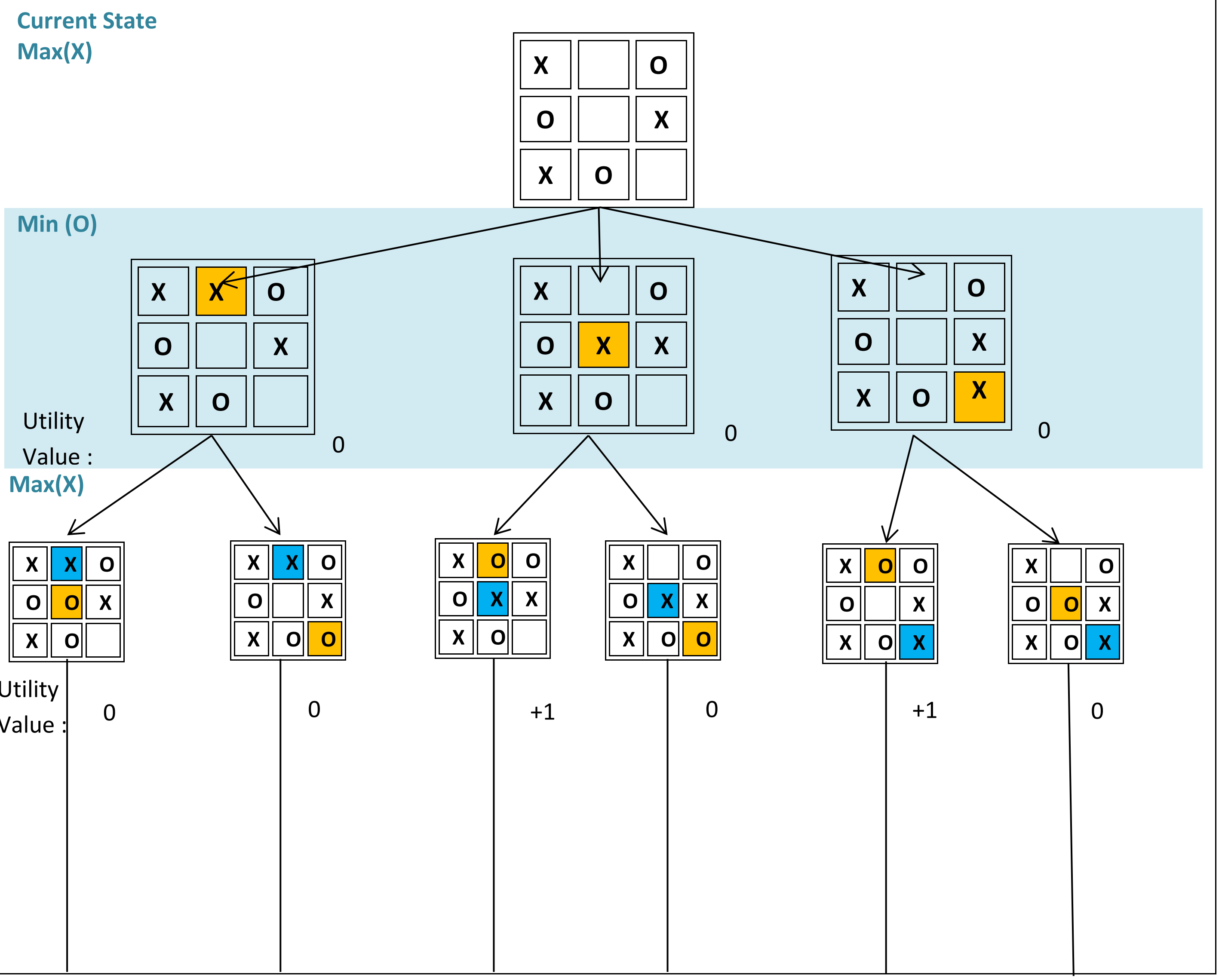
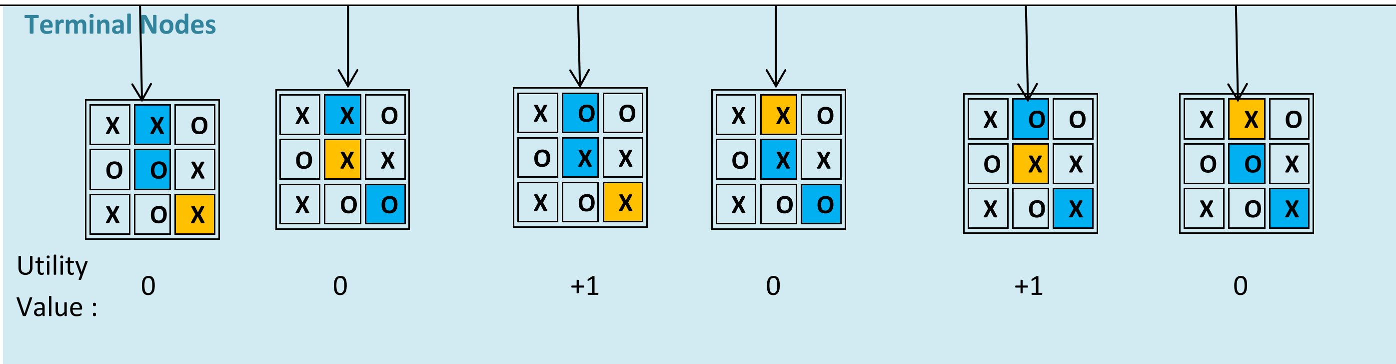


Figure 1. A tic-tac-toe board state.

Solution:



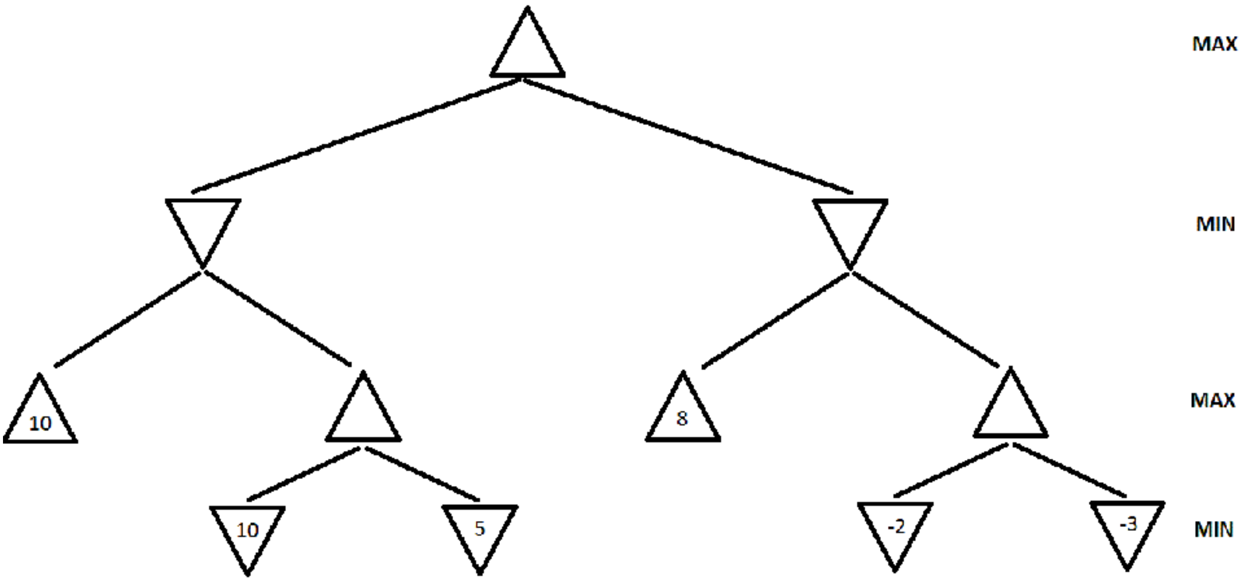


Blocks played already

Block played in the current turn

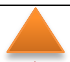
The Minimax algorithm would choose to proceed with the paths where the maximum utility value is achieved for the Max user. However in this scenario, the Max player will have all the paths with a utility value of 0 and has equal weightage in choosing the paths.

Task 2

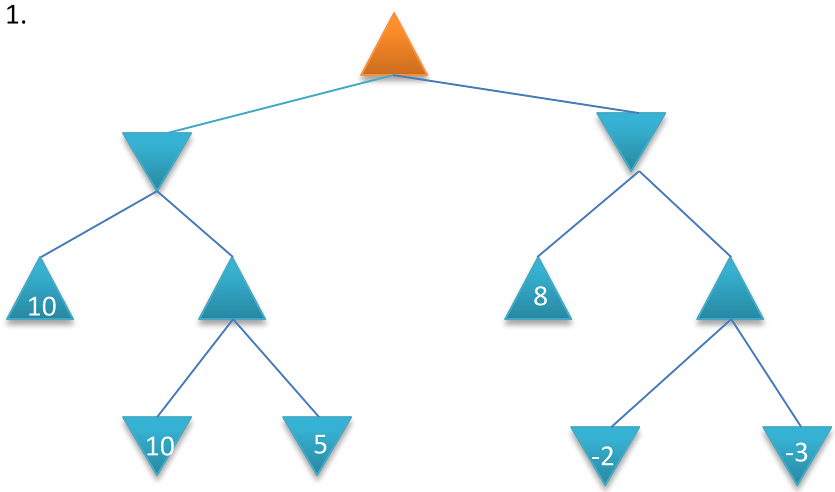
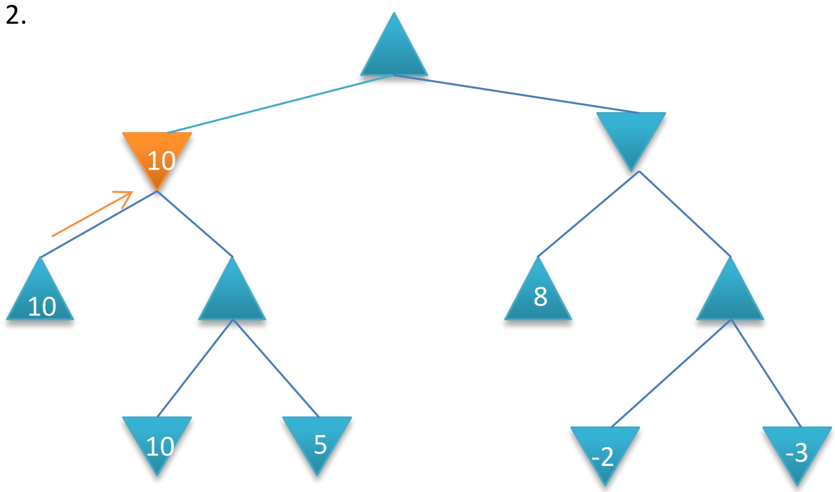
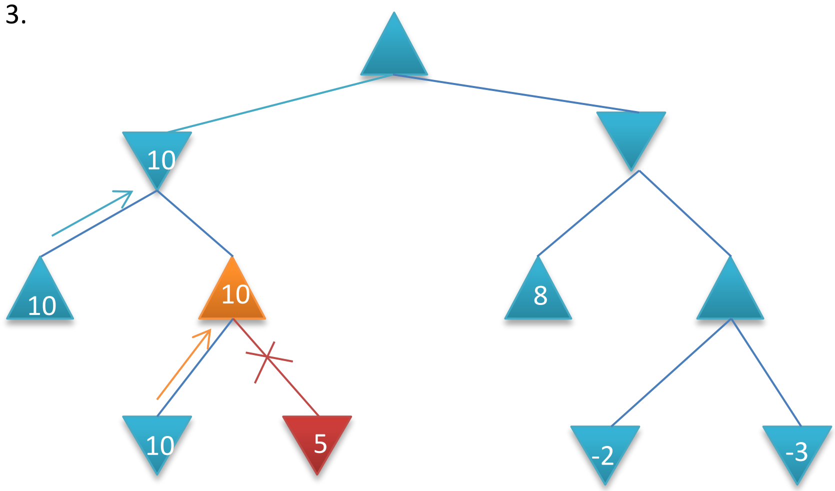


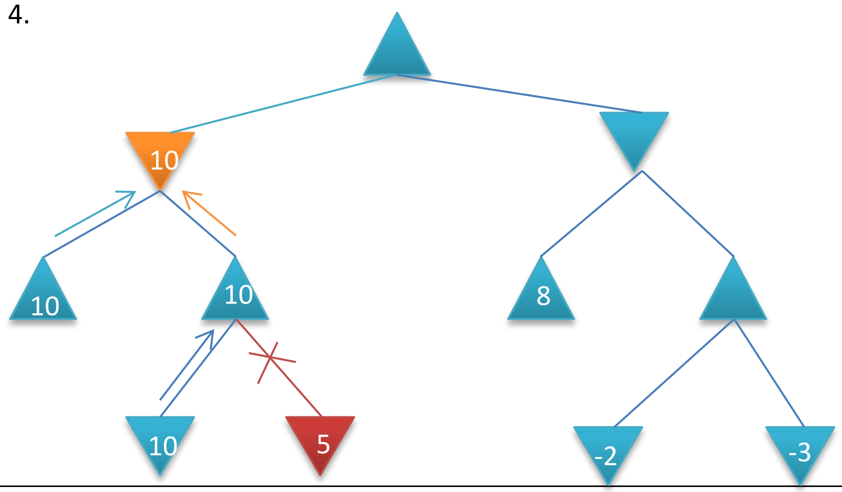
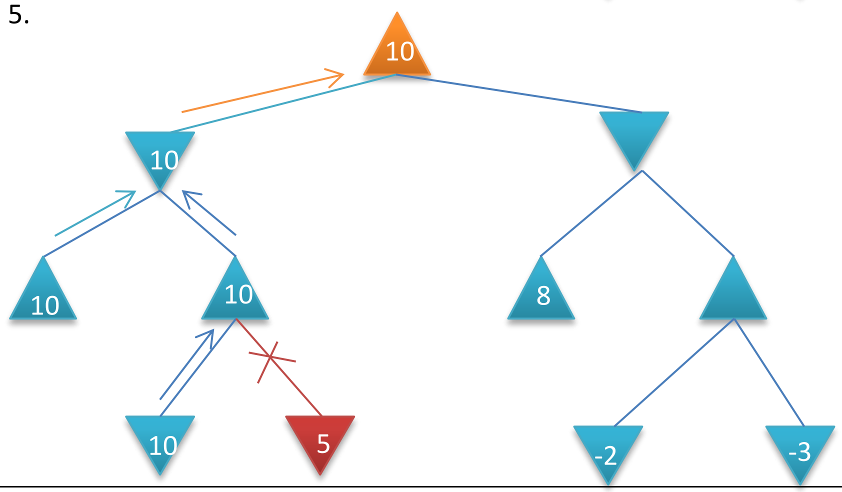
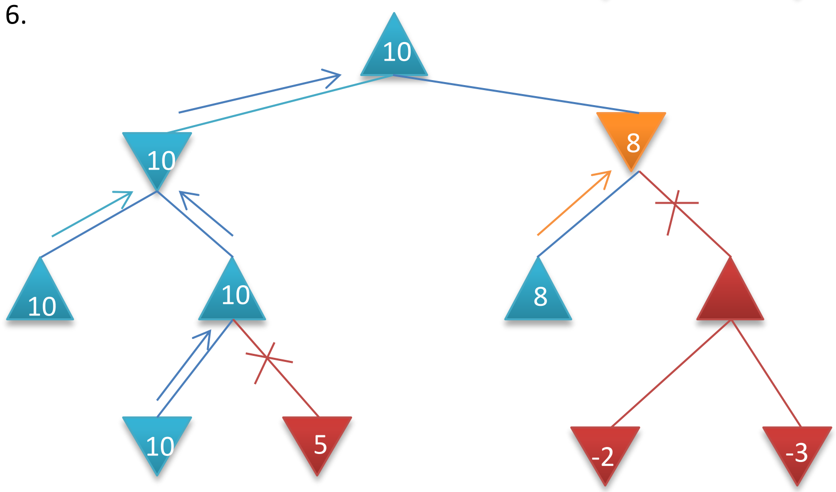
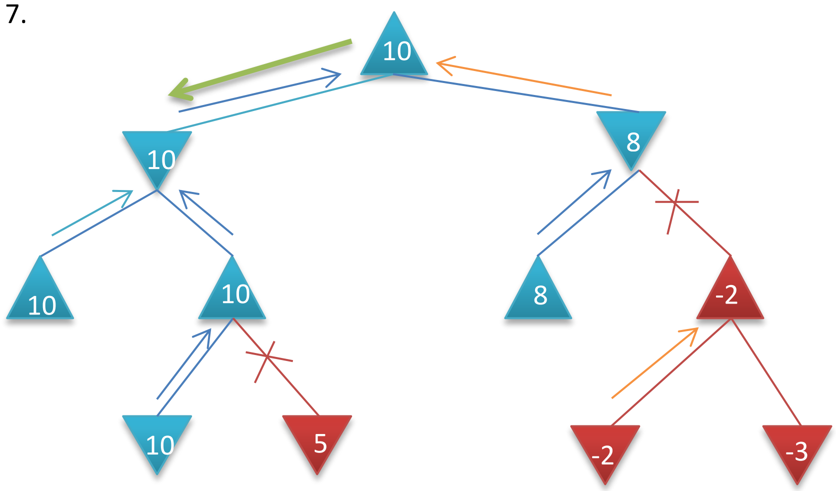
a. **(4308: 10 points, 5360: 10 points) In the game search tree of Figure 2, indicate what nodes will be pruned using alpha-beta search, and what the estimated utility values are for the rest of the nodes. Assume that, when given a choice, alpha-beta search expands nodes in a left-to-right order. Also, assume the MAX player plays first. Finally indicate which action the Minimax algorithm will pick to execute.**

Min	**Max**
If v <= α Then prune everything Else if V < β β = v continue	If V >= β Then prune everything Else if v > α α = v continue

 - Node currently under evaluation

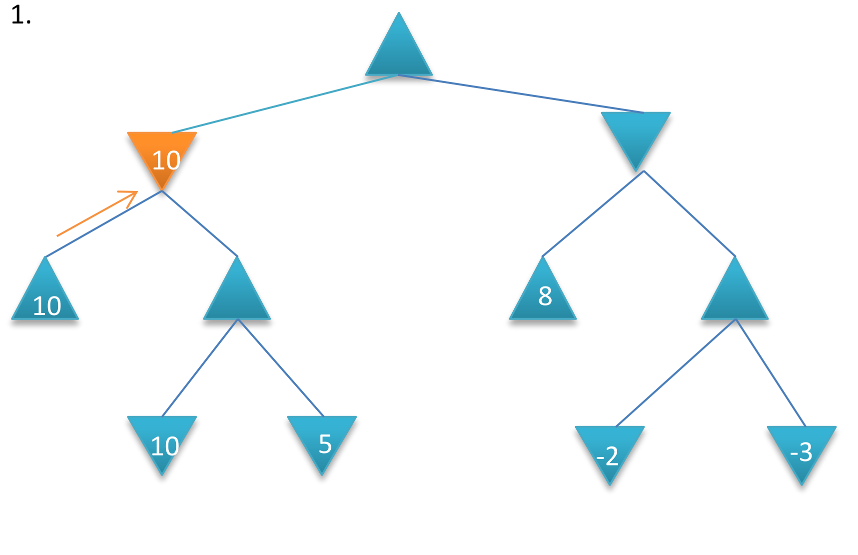
 - Pruned node

1. 	V = -∞ α = -∞ β = +∞
2. 	As per the problem statement, alpha beta search expands the node from left to right. Taking the left most node, Value [-∞,∞] is passed down from the Max node V = ~~+∞~~10 (Value from the terminal node) α = -∞ β = +∞ is 10<= -∞? No is 10 < ∞ ? Yes So β = 10 [-∞, 10]
3. 	Expanding the next node under the Min node, Value [-∞,10] is passed down from the Max node V = 10 (Value from the terminal node) α = -∞ β = 10 is 10>=10? Yes Prune the remaining nodes

4. 	Current value $[-\infty,10]$ V = 10 (Value from the successor node) $\alpha = -\infty$ $\beta = 10$ is $10 <= -\infty$? No is 10 < 10 ? No so, continue $[-\infty,10]$
5. 	Current value $[-\infty,\infty]$ V = 10 (Value from the successor node) Is $10>= \infty$? No Is $10> -\infty$? Yes So, $\alpha = 10$ $[10, \infty]$
6. 	Value $[10,\infty]$ is passed down from the Max node V= 8 (Value from the successor node) $\alpha = 10$ $\beta = \infty$ is $8<= 10$? Yes So, Prune the remaining nodes
7. 	Finding the maximum of 10 and 8 to give value to the Max node : 10 **Estimated Utility value** for the Pruned Max node (-2) is found by finding the Max value of its child nodes. Estimated Utility value of other nodes are presented in the imager The node that would be chosen by the Minmax algorithm would be node 10, the path is marked by a green arrow.

b. (4308: 5 points, 5360: 5 points) This question is also on the game search tree of Figure 2. Suppose we are given some additional knowledge aboutthe game: the maximum utility value is 10, i.e., it is not mathematically possible for the MAX player to get an outcome greater than 10. How can thisknowledge be used to further improve the efficiency of alpha-beta search? Indicate the nodes that will be pruned using this improvement. Again,assume that, when given a choice, alpha-beta search expands nodes in a left-to-right order, and that the MAX player plays first.

1.



As per the problem statement, alpha beta search expands the node from left to right. Taking the left most node,

Value [-∞,∞] is passed down from the Max node

V = ~~+∞~~10 (Value from the terminal node)

α = -∞

β = +∞

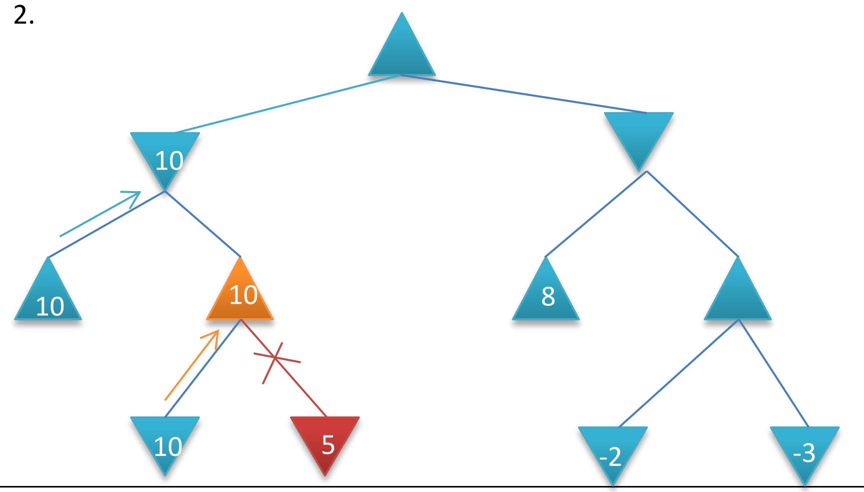
is 10<= -∞? No

is 10 < ∞ ? Yes

So β = 10

[-∞, 10]

2.



Expanding the next node under the Min node,

Value [-∞,10] is passed down from the Max node

V = 10 (Value from the terminal node)

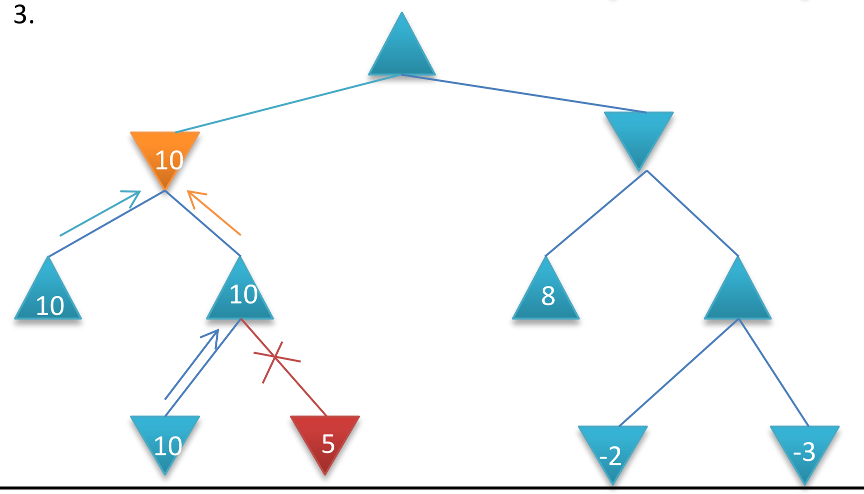
α = -∞

β = 10

is 10>=10? Yes

Prune the remaining nodes

3.



Current value [-∞,10]

V = 10 (Value from the successor node)

α = -∞

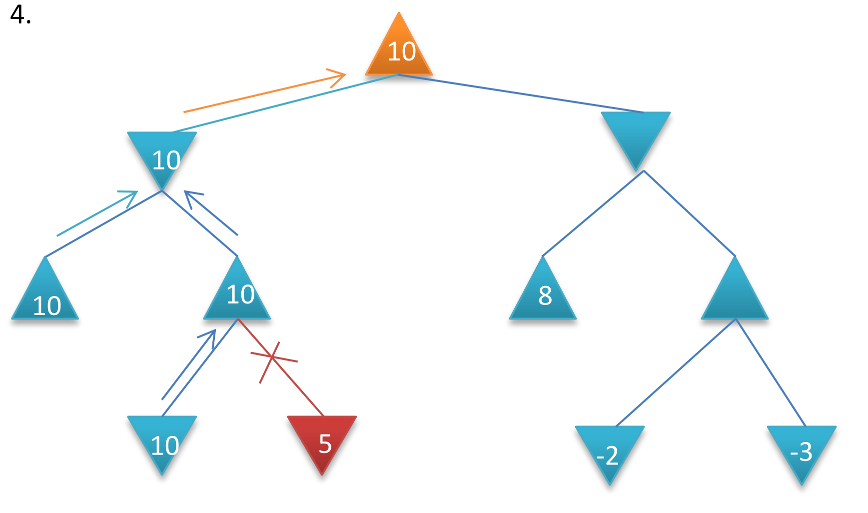
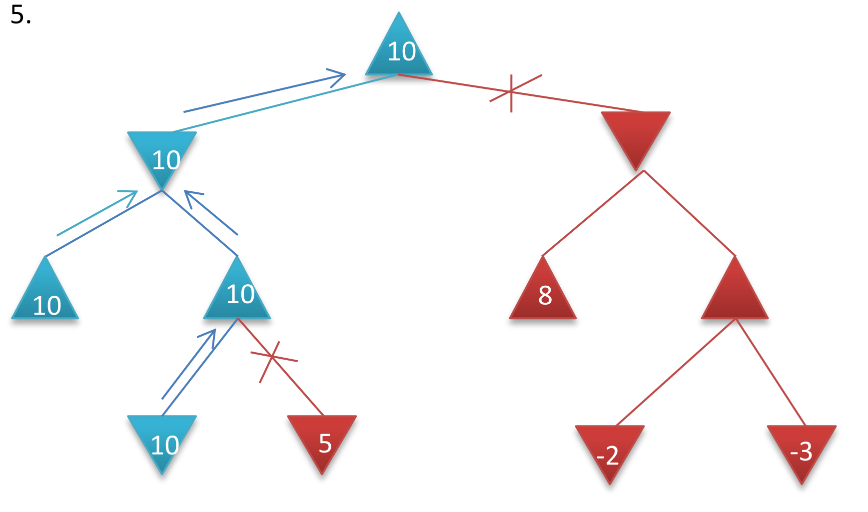
β = 10

is 10<= -∞? No

is 10 < 10 ? No

so, continue

[-∞,10]

4. 	Current value [-∞,∞] V = 10 (Value from the successor node) Is 10>= ∞? No Is 10> -∞? Yes So, α = 10 [10, ∞]
5. 	Since the problem statement mentions that there is an additional knowledge that the max utility value is 10, and that it is reached through an already explored path, the remaining nodes can be pruned without expansion. With this additional information, the number of steps involved in finding the path to the Max utility value is achieved in lesser number of steps.

Task 3

Suppose that you want to implement an algorithm that will compete on a two-player deterministic game of perfect information. Your opponent is asupercomputer called DeepGreen. DeepGreen does not use Minimax. You are given a library function DeepGreenMove(S), that takes any state S as an argument, and returns the move that DeepGreen will choose for that state S (more precisely, DeepGreenMove (S) returns the state resulting fromthe opponent's move).

Write an algorithm in pseudocode (following the style of the Minimax pseudocode) that will always make an optimal decision given the knowledge we have about DeepGreen. You are free to use the library function DeepGreenMove(S) in your pseudocode. What advantage would this algorithm have over Minimax? (if none, Justify).

```
function Minimax-Decision(state) returns an action
inputs: state, current state in game
return the a in Actions(state) maximizing Min-Value(Result(a, state))

function Max-Value(state) returns a utility value
if Terminal-Test(state) then return Utility(state)
v ← -∞
for a, s in Successors(state) do v Max(v, Min-Value(s))
return v

function Min-Value(state) returns a utility value
if Terminal-Test(state) then return Utility(state)
v ←∞
```

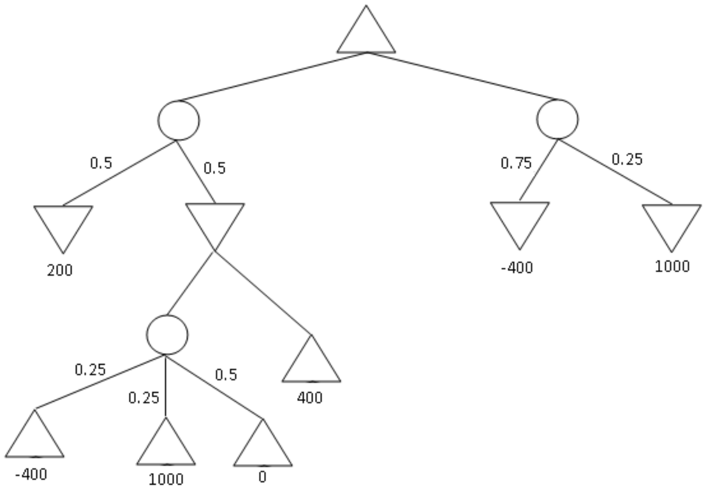
for each a in ACTIONS(state) do
v ←MIN(v, MAX-VALUE(RESULTS(s, a)))
if v <= UTILITY(DeepGreenMove(state))
then return MAX(v, DeepGreenMove(state))
else
return v

Advantage of the modified algorithm with DeepGreenMove library over Minmax algorithm

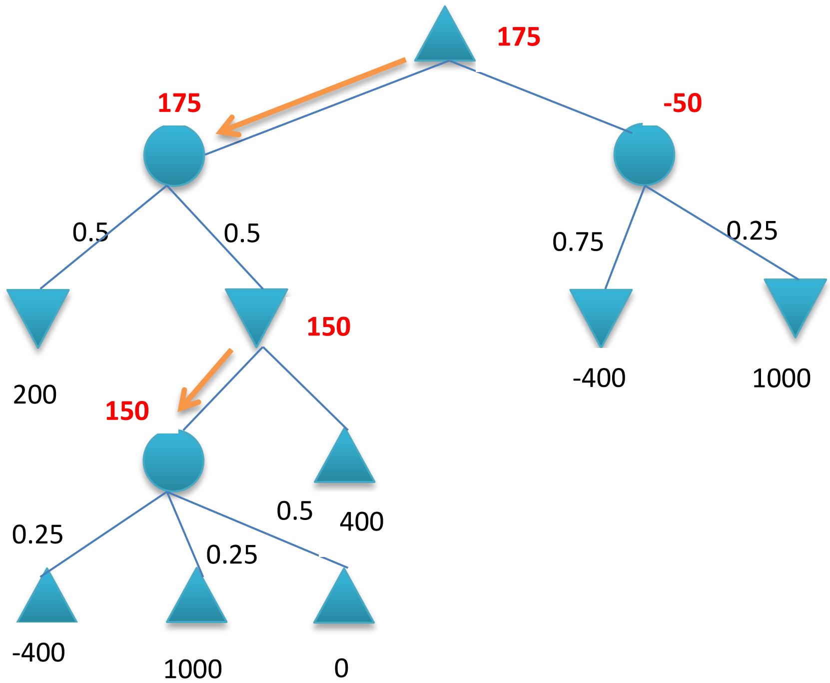
In case of an optimal opponent as a Min player, the Minmax algorithm behaves the same way as the algorithm with the DeepGreenMove library calls. Since, the moves made by the optimal opponent and the Deep Green Move are the same and that the Minmax algorithm works under an assumption that the opponent is optimal.
In case of a sub optimal opponent, the modified algorithm with DeepGreenMove has the advantage of providing a better utility value. Wherein the maximum utility value may be present in a different path which the optimal player may end of leading to lower utility value. However, with a sub optimal player and the Deep Green move library, the Max player has the chances to reach a better utility value..

Task 4

Find the value of every non-terminal node in the expectiminmax tree given above. Also indicate which action will be performed by the algoirithm.What is lowest and highest possible outcome of a single game if the minmax strategy is followed.



Value found for every non terminal node is marked in red. Action that will be performed by the algorithm is marked in the orange arrow.



Action performed by the algorithm,

1. The top Max node would choose the path along 175 (between 175 and -50)
2. There are equal chances that the Min player would end up in ether the node with 200 or the node with 150.
3. If the node with 150 is reached, any optimal min player would choose the chance node 150 (between 150 and 400)
4. This may lead the Max player to end with a terminal node value of -400 or 1000 or 0

Highest Possible Outcome of a single game : 1000
Lowest Possible outcome of a single game : -400

Task 5

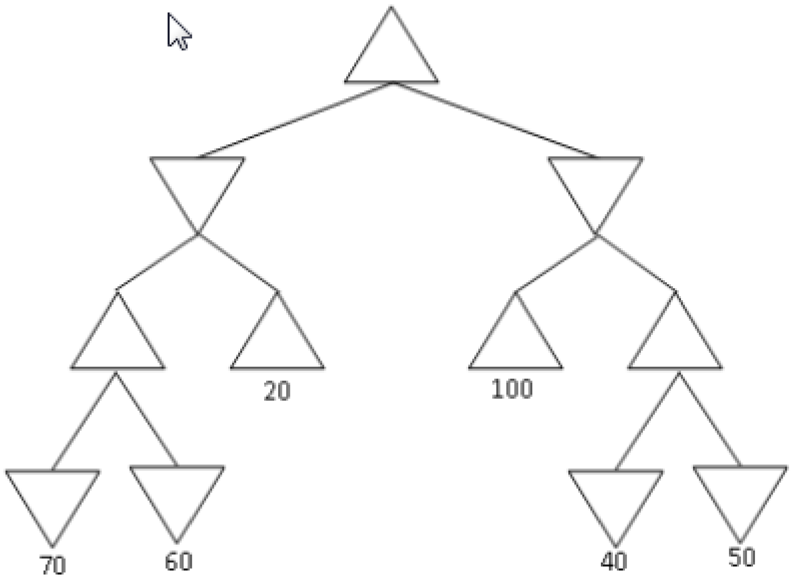
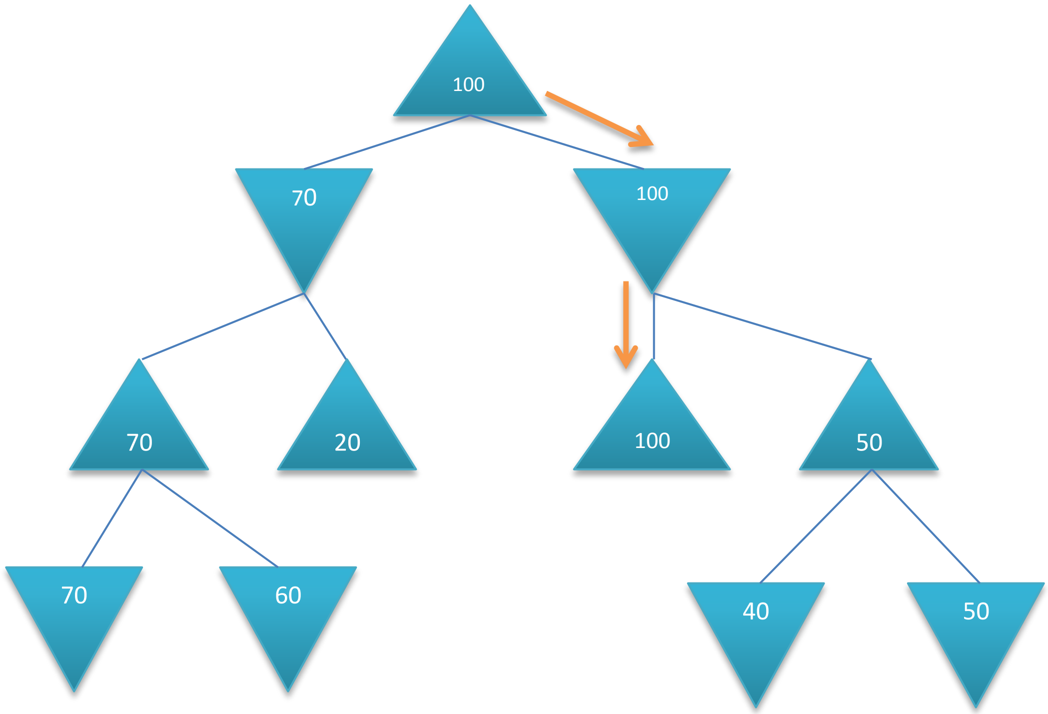


Figure 4: Yet another game search tree

Consider the MINIMAX tree above. Suppose that we are the MAX player, and we follow the MINIMAX algorithm to play a full game against anopponent. However,
we do not know what algorithm the opponent uses
.
Under these conditions, what is the best possible outcome of playing the full game for the MAX player? What is the worst possible outcome for theMAX player? Justify your answer.
NOTE: the question is not asking you about what MINIMAX will compute for the start node. It is asking you what is the best and worst outcome ofa complete game under the assumptions stated above.

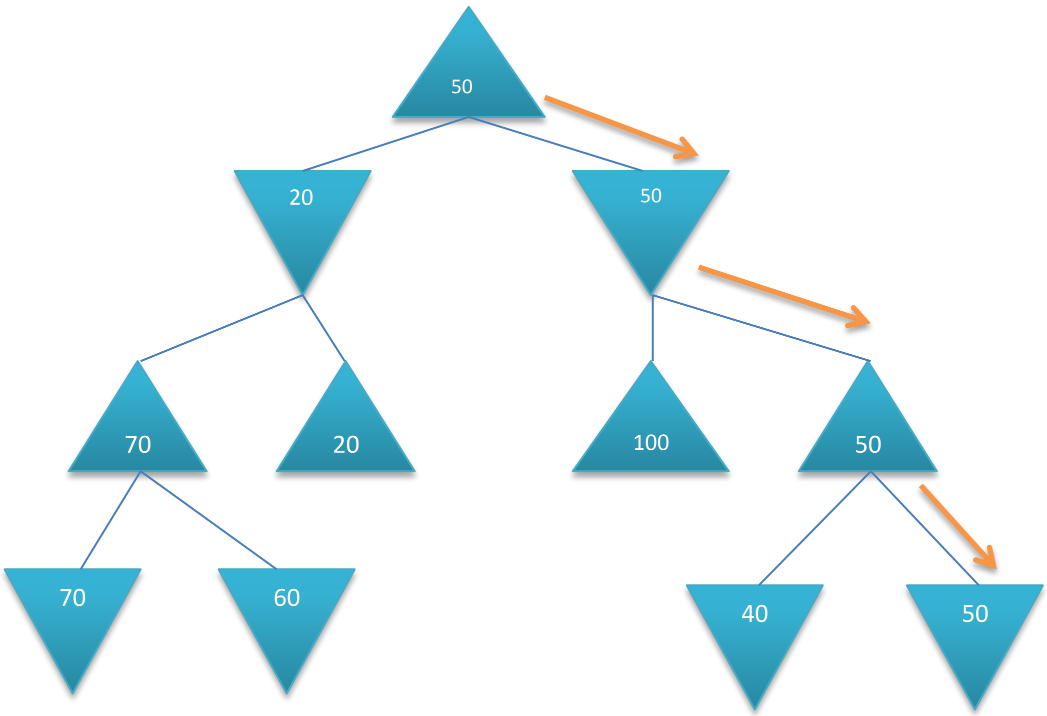
The Minmax algorithm would always choose the move that would yield the maximum utility value. Hence in the Nodes with 70 and 60, the algorithm would always choose 70 for the Max node and similarly, 50 would be chosen among 50 and 40.

Best Case Scenario



The Base outcome of the game would be for the Max player to arrive at the node with a utility value of 100.
This is under the assumption that the Min player would not be making the optimal move and choosing the node with a higher utility value.
In such case, the Max player would be able to achieve the best outcome of utility value 100.

Worst Case Scenario



In case of an optimal opponent as a Min player who would always reach for the best move possible, then the possible outcome for the Max player would be 50.

This happens when the Min player chooses the node 50 where in the Max player chooses the Max value among 40 and 50.

Task 6

Two logical statements S1 and S2 are logically equivalent if (S1 <=> S2) is valid. We have two knowledge bases, KB1 and KB2. Write a function CHECK_EQUIVALENCE(KB1, KB2) that:
returns true if KB1 and KB2 are logically equivalent.
returns false otherwise.
Your pseudocode can re-use any code from the textbook or slides, and can call any of the functions given in the textbook or slides, as long as such code and functions are used correctly, with correct names for the functions, and with well-specified values for all variables and arguments.

Truth table for Biconditional equivalence S1 ⇔ S2 is as follows,

S1	S2	S1⇔S2
True	True	True
True	False	False
False	True	False
False	False	True

So, the pseudo code should return True only when the model value of both S1 and S2 are same.

```
Function CHECK_EQUIVALENCE(KB1, KB2)
   Inputs: KB1 – Knowledge Base 1, a sentence in propositional logic
           KB2 – Knowledge Base 2, a sentence in propositional logic
   Symbols ← a list of the propositional symbols in KB1 and KB2
   Return CHECK_ALL(KB1, KB2, symbols, [])

Function CHECK_ALL(KB1, KB2, symbols, model) returns true or false
   If Empty?(symbols) then
      If (PL-TRUE?(KB1, model) and PL-TRUE?(KB2, model) ) then return true
      If (PL-FALSE?(KB1, model) and PL-FALSE?(KB2, model) ) then return true
      else return false
   else
      do
       P←FIRST(symbols)
      rest←REST(symbols)
      return (CHECK_ALL (KB1, KB2, rest, EXTEND(P, true, model))
              and CHECK_ALL (KB1, KB2, rest, EXTEND(P, false, model))
              and CHECK_ALL (KB2, KB1, rest, EXTEND(P, true, model))
              and CHECK_ALL (KB2, KB1, rest, EXTEND(P, false, model))
```

PL-TRUE returns true if the sentence holds true within the model
PL-FALSE returns true if the sentence does not true within the model
EXTEND (p, true, model) returns a partial new model in which p has a value true.

Task 7

A	B	C	KB	S1
True	True	True	True	True
True	True	False	False	True
True	False	True	True	True
True	False	False	False	True
False	True	True	False	False
False	True	False	False	False
False	False	True	True	True
False	False	False	False	False

KB and S1 are two propositional logic statements, that are constructed using symbols A, B, C, and using various

connectives. The above truth table shows, for each combination of values of A, B, C, whether KB and S1 are true or false.

Part a: Given the above information, does KB entail S1? Justify your answer

Definition of Entailment : Knowledge base KB entails sentence α, if and only if α is true in all worlds where KB is true

So, to ensure KB entails S1, we need to see if S1 is true for all cases where KB is true.
In the truth table, marking all scenarios where KS is true.

A	B	C	KB	S1
True	True	True	True	True
True	True	False	False	True
True	False	True	True	True
True	False	False	False	True
False	True	True	False	False
False	True	False	False	False
False	False	True	True	True
False	False	False	False	False

In all the marked models, where KB is true, S1 also is true. Hence we can say that **KB entails S1**.

Part b: Given the above information, does statement NOT(KB) entail statement NOT(S1)? Justify your answer.

To verify if NOT(KB) entails NOT(S1), adding additional columns in the above table and marking the models where Not(KB) holds true,

A	B	C	KB	NOT(KB)	S1	NOT(S1)
True	True	True	True	False	True	False
True	True	False	False	True	True	False
True	False	True	True	False	True	False
True	False	False	False	True	True	False
False	True	True	False	True	False	True
False	True	False	False	True	False	True
False	False	True	True	False	True	False
False	False	False	False	True	False	True

We can see that not all the models where NOT(KB) is true is true for NOT(S1). Models [(A = True, B= True, C=False) and (A = True, B= False, C=False)] are true for NOT(KB) but are false for NOT(S1), hence, **NOT(KB) does not entail NOT(S1)**

Task 8

Suppose that some knowledge base contains various propositional-logic sentences that utilize symbols A, B, C, D (connected with variousconnectives). There are only two cases when the knowledge base is false:
- First case: when A is true, B is false, C is true, D is true.
- Second case: when A is false, B is false, C is true, D is false.
In all other cases, the knowledge base is true. Write a conjunctive normal form (CNF) for the knowledge base.

Building the truth table based on the provided information,

A	B	C	D	KB
True	True	True	True	True
True	True	True	False	True
True	True	False	True	True
True	True	False	False	True

True	False	True	True	False
True	False	True	False	True
True	False	False	True	True
True	False	False	False	True
False	True	True	True	True
False	True	True	False	True
False	True	False	True	True
False	True	False	False	True
False	False	True	True	True
False	False	True	False	False
False	False	False	True	True
False	False	False	False	True

The rows marked in blue are the ones that have a false value. So, to create a Conjunctive normal form,

Row 5 = $A \wedge \neg B \wedge C \wedge D$
Row 14 = $\neg A \wedge \neg B \wedge C \wedge \neg D$

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

Applying De Morgans law, $\neg (\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$

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

Applying double negation elimination, $\neg(\neg\alpha) \equiv \alpha$

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

This is the CNF of the knowledge base where there is a conjunction of, disjunction of literals.

Task 9

Consider the KB
A => B
B <=> C
D => A
E => D
C AND E => F
E

Show that this entails F by
i. Forward Chaining

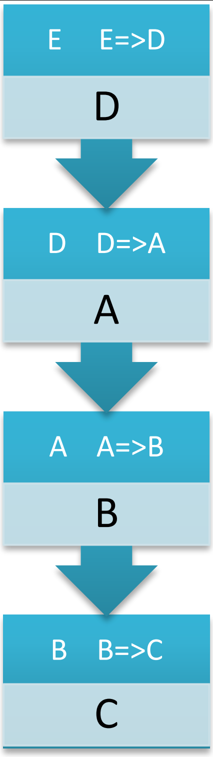
In the given knowledge base, simplifying B⇔C to (B=> C) ^ (C=>B)

So, the horn form of the knowledge base would be,

(A=> B) ^ (B=>C) ^ (C=>B) ^ (D=>A) ^ (E=>D) ^ (C^E =>F) ^ E

Since we know E is true, starting with E,

Applying Modus Ponens for the Horn Form,



So, from the above inference from Modus Ponens, we know that D, A, B and C are true.

From the knowledge base, we also know that E is true.

The knowledge base also has an entry that C^E => F. We now know that C and E are true, which means F is true for this knowledge base, hence the given knowledge base entails the α value, F.

ii. Backward Chaining

In the given knowledge base, simplifying B⇔C to (B=> C) ^ (C=>B)

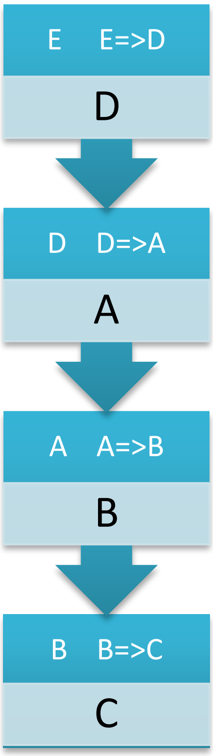
So, the horn form of the knowledge base would be,

(A=> B) ^ (B=>C) ^ (C=>B) ^ (D=>A) ^ (E=>D) ^ (C^E =>F) ^ E

As per backward chaining, building the stack from the goal level F.

~~E~~	• As per KB, for D to be true, E should be true.. E is true in the KB.
D	• As per KB, for A to be true, D should be true. D=>A
A	• As per KB, for B to be true, A should be true, A=>B
B	• As per KB, for C to be true, B should be true. B=>C
~~E~~	• As per KB, C^E => F • So, E and C should be true. E is true in the KB
C	• As per KB, C^E => F • So, C and E should be true
F	• Goal

Since E is true in the KB, applying Modus Ponens for elements from the stack,



Now, with the above inference, C is also true. Hence proving that the Knowledge base entails F since both C and E are true in the knowledge base.

iii. Resolution

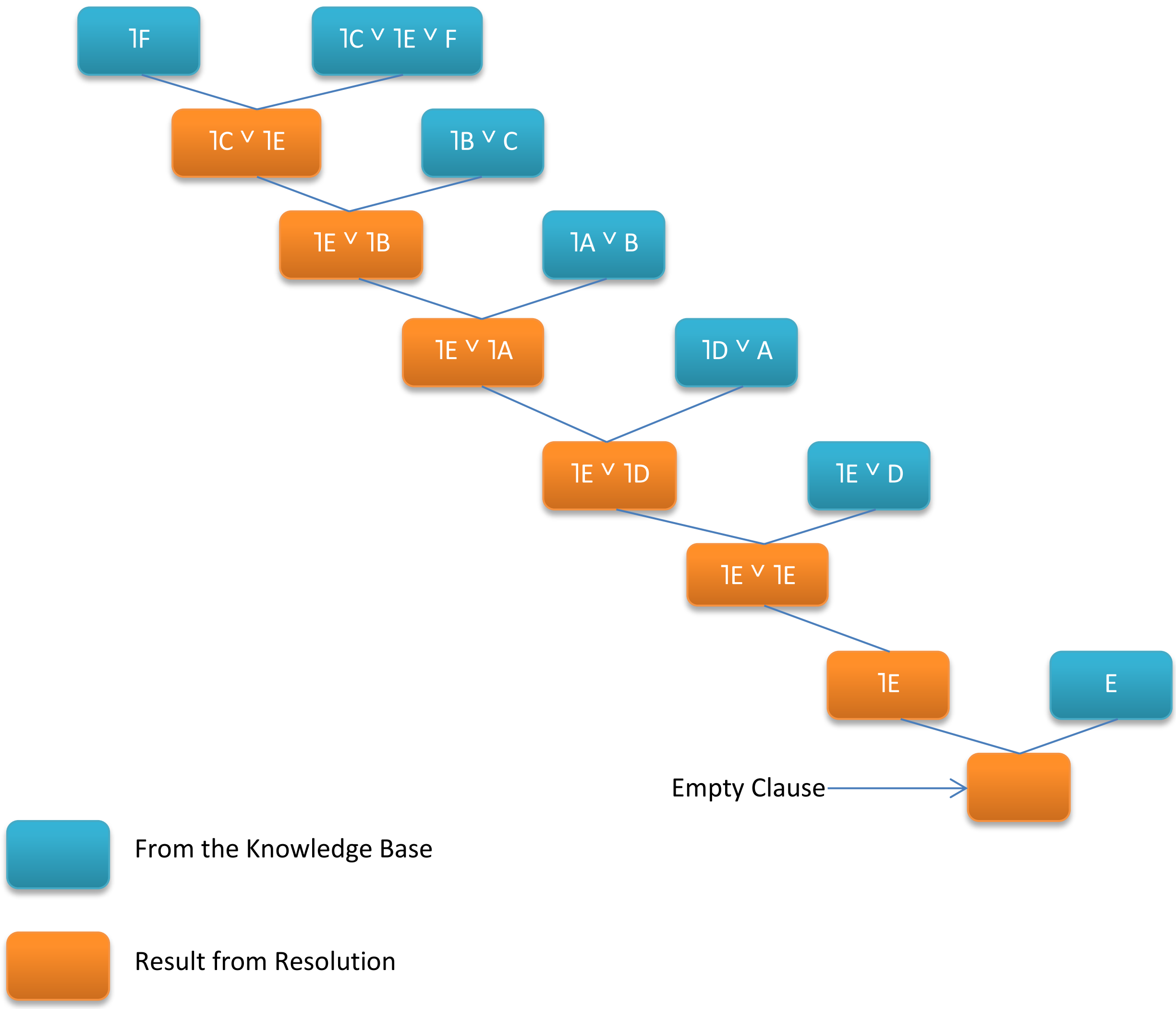
Given KB is
A => B
B <=> C
D => A
E => D
C AND E => F
E

α is F

To implement Resolution, rewriting the KB into Conjunctive normal form,

A=>B	¬A ∨ B	Implication Elimination
B⇔C	(B=>C) ^ (C=>B)	Biconditional Elimination
	(¬B ∨ C) ^ (¬C ∨ B)	Implication Elimination
D =>A	¬D ∨ A	Implication Elimination
E =>D	¬E ∨ D	Implication Elimination
C ^ E => F	¬(C ^ E) ∨ F	Implication Elimination
	¬C ∨ ¬E ∨ F	De Morgans
E	E	

Since α is F, the literal to be started with, is ¬F



Hence, KB and α are unsatisfied. So, KB entails α, as per the resolution..

Task 10

In April, John and Mary sign the following contract:
- If it rains in May, then John must give Mary a check for $10,000
- If John gives Mary a check for $10,000, Mary must mow the lawn.
What truly happened those days is the following:
- It did not rain in May.
- John gave Mary a check for $10,000
- Mary mowed the lawn.

Part a: Write a first order logic statement to express the contract. Make sure that you clearly define what constants and predicates that you use are.(NOTE: DO NOT use functions)

In the contract above, the constants are
John
Mary
May

Predicates that will be used are,
Rain(May)
Give10KCheque(John, Mary)

MowTheLawn(Mary)

So as per the contract statement,

If it rains in May, then John must give Mary a check for $10000	Rain(May) => Give10KCheque (John, Mary)
If John gives Mary a check for $10,000, Mary must mow the lawn.	Give10KCheque (John, Mary)=> MowTheLawn(Mary)

First Order Logic statement to express the contract,
(Rain(May) => Give10KCheque (John, Mary)) ^ (Give10KCheque (John, Mary)=> MowTheLawn(Mary))

Part b: Write a logical statement to express what truly happened. When possible, use the same predicates and constants as in question 6a. If you needto define any new predicates or constants, clearly define what they stand for.

Constants used in the logic expressions are,
John, Mary and May

Predicates that will be used are,
Rain(May)
Give10KCheque (John, Mary)
MowTheLawn(Mary)

Actual happening,

- It did not rain in May.	˥(Rain(May))
- John gave Mary a check for $10,000	Give10KCheque (John, Mary)
- Mary mowed the lawn.	MowTheLawn(Mary)

Logic statement for actual happening,
˥(Rain(May)) ^ Give10KCheque (John, Mary) ^ MowTheLawn(Mary)

Part c: Define the symbols required to convert any KB involved in the above domanin from FOL to Propositional logic (Your symbols must allowme to convert ANY KB that uses the predicates and constants as decribed previously).

Symbols required to convert the previously used Predicates and constants are as follows

Constants	Symbols
April	T1
May	T2
John	P1
Mary	P2

Predicates	Symbols
Rain(April)	R1
Rain(May)	R2
Rain(John)	R3
Rain(Mary)	R4
Give10KCheque (John, John)	G1
Give10KCheque (John, Mary)	G2
Give10KCheque (Mary, John)	G3
Give10KCheque (Mary, Mary)	G4
Give10KCheque (John, April)	G5

Give10KCheque (John, May)	G6
Give10KCheque (Mary, April)	G7
Give10KCheque (Mary, May)	G8
Give10KCheque (April, John)	G9
Give10KCheque (April, Mary)	G10
Give10KCheque (May, John)	G11
Give10KCheque (May, Mary)	G12
Give10KCheque (April, April)	G13
Give10KCheque (April, May)	G14
Give10KCheque (May, April)	G15
Give10KCheque (May, May)	G16
MowTheLawn(John)	M1
MowTheLawn(Mary)	M2
MowTheLawn(April)	M3
MowTheLawn(May)	M4

Part d: Use the sybols given in part c, to convert the answers to part a and b to Propositional Logic.

Using the symbols from Part c,

First order logic for the contract is
(Rain(May) => Give10KCheque (John, Mary)) ^ (Give10KCheque (John, Mary)=> MowTheLawn(Mary))

The propositional Logic for the contract is,
(R2 => G2) ^ (G2 => M2)

First order logic for the true happening is,
˥(Rain(May)) ^ Give10KCheque (John, Mary) ^ MowTheLawn(Mary)

The propositional Logic for the true happening is,
˥R2 ^ G2 ^ M2

Part e: Was the contract violated or not, Justify your answer (Note: if the sequence of events that occured entails the contract then it was not violated)

As per the contract, if a specific condition is met Give10Kcheque(John, Mary) should be true and if another condition is true, MowTheLawn(Mary) should be true.

As per the true happening, Give10Kcheque(John, Mary) is true and MowTheLawn(Mary) is true. Hence the implications have happened in the actual case.

As per the definition provided, the sequence of the actually happened events is a subset of the contract. Hence the actual sequence of events entails the contract and hence the contract is not violated.

Reference :

Slide Chapter 7: Logical Agents for the pseudocode of TT-ENTAILS