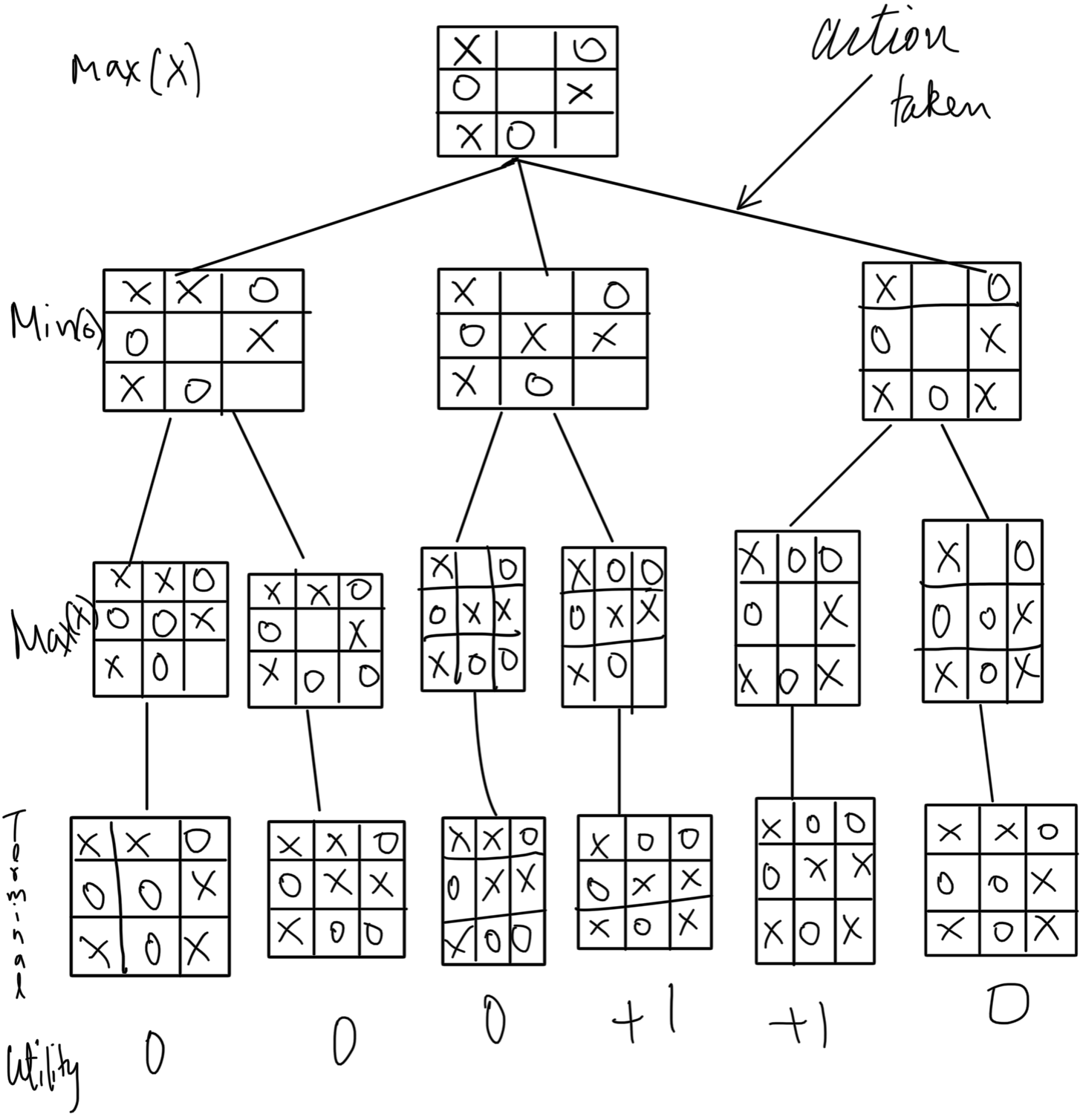
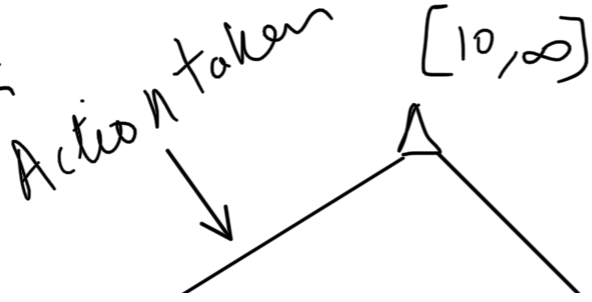
Homework 3

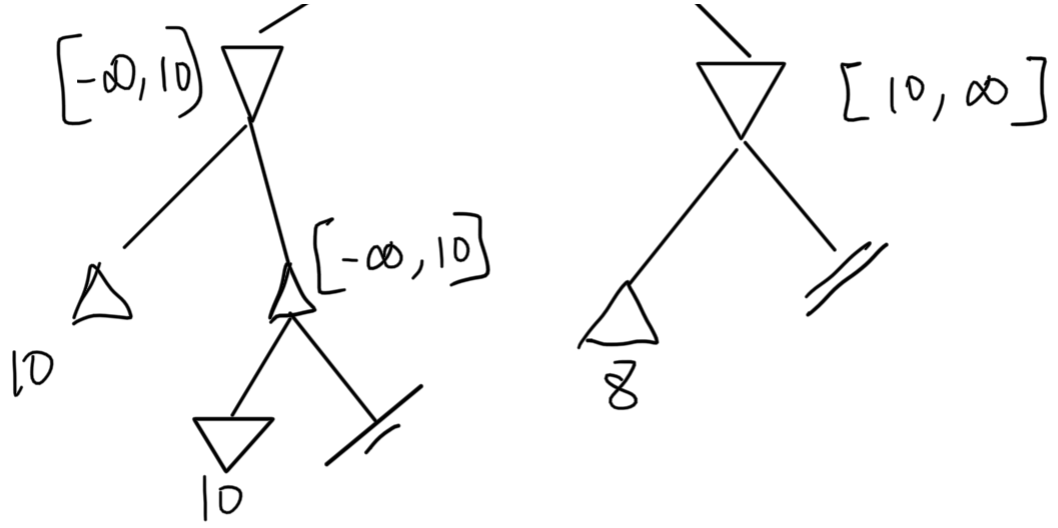
Task - 1



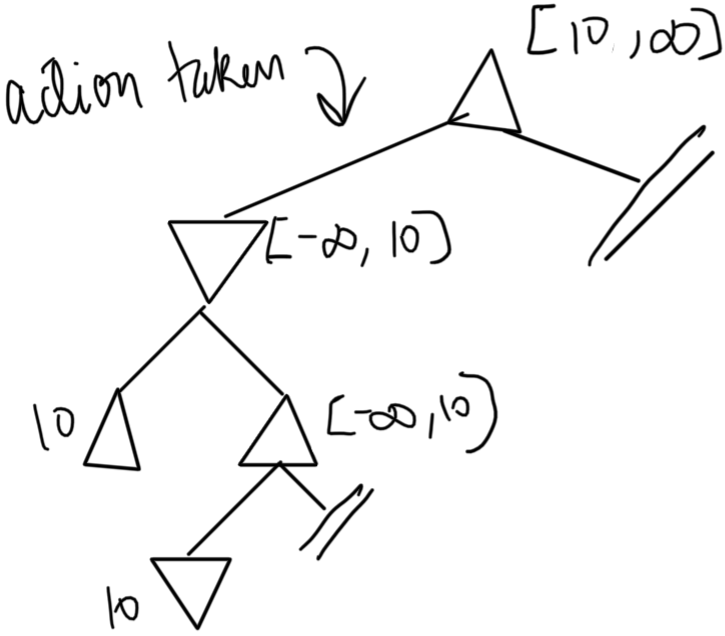
Task 2

a)





b) If we are given the max utility value is 10, then we can prune any branch after it finds a successor with a value of 10.



Task 3

Since this is a deterministic game, we will know what move DeepGreen would return. We can optimize the code by not taking all

the successors for a min node, but rather take a single successor for each node.

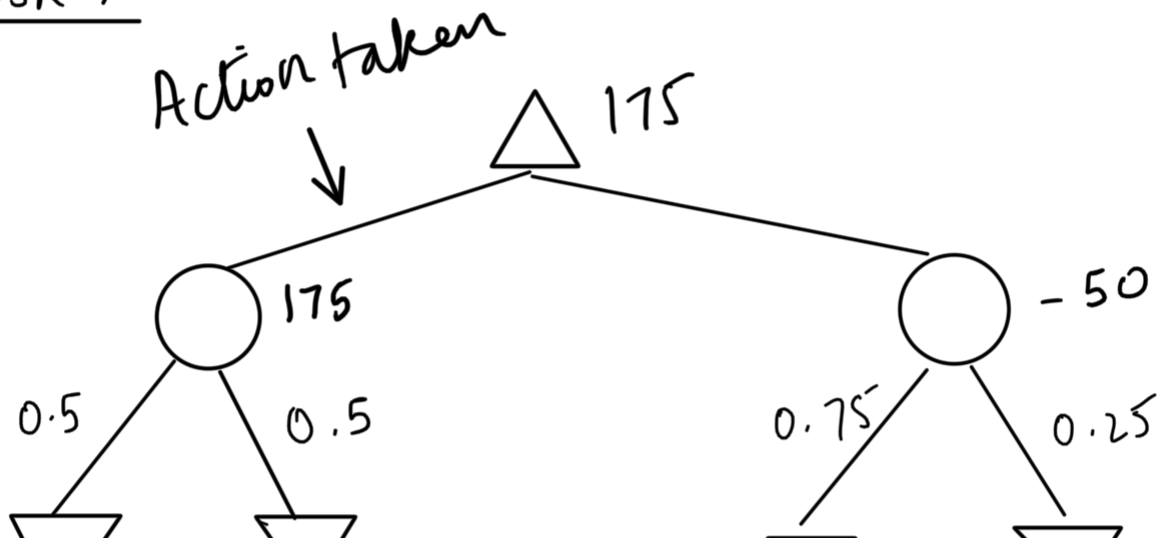
Pseudo code:

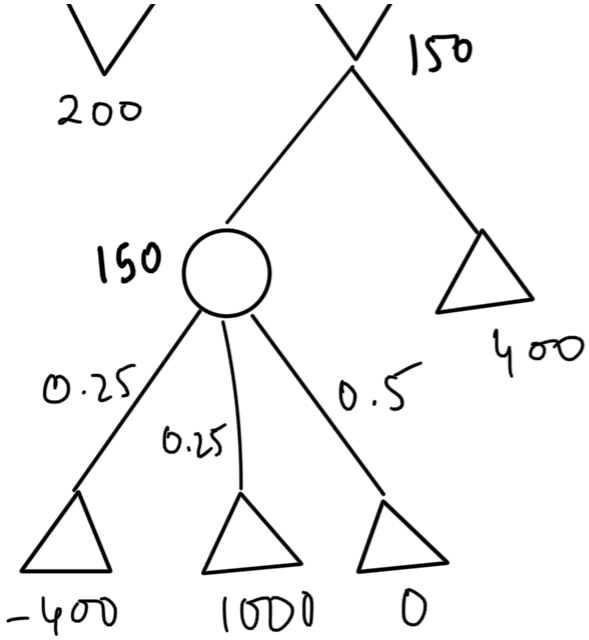
Minimax-decision and max-value function remain the same.

function MIN-VALUE (state) returns a utility value
if Terminal-Test (state) then return UTILITY (state)
return MAX-VALUE (DeepGreenMove (State))

The advantage this algorithm will have is that it will explore fewer nodes as it will only return one successor for min nodes.

Task 4

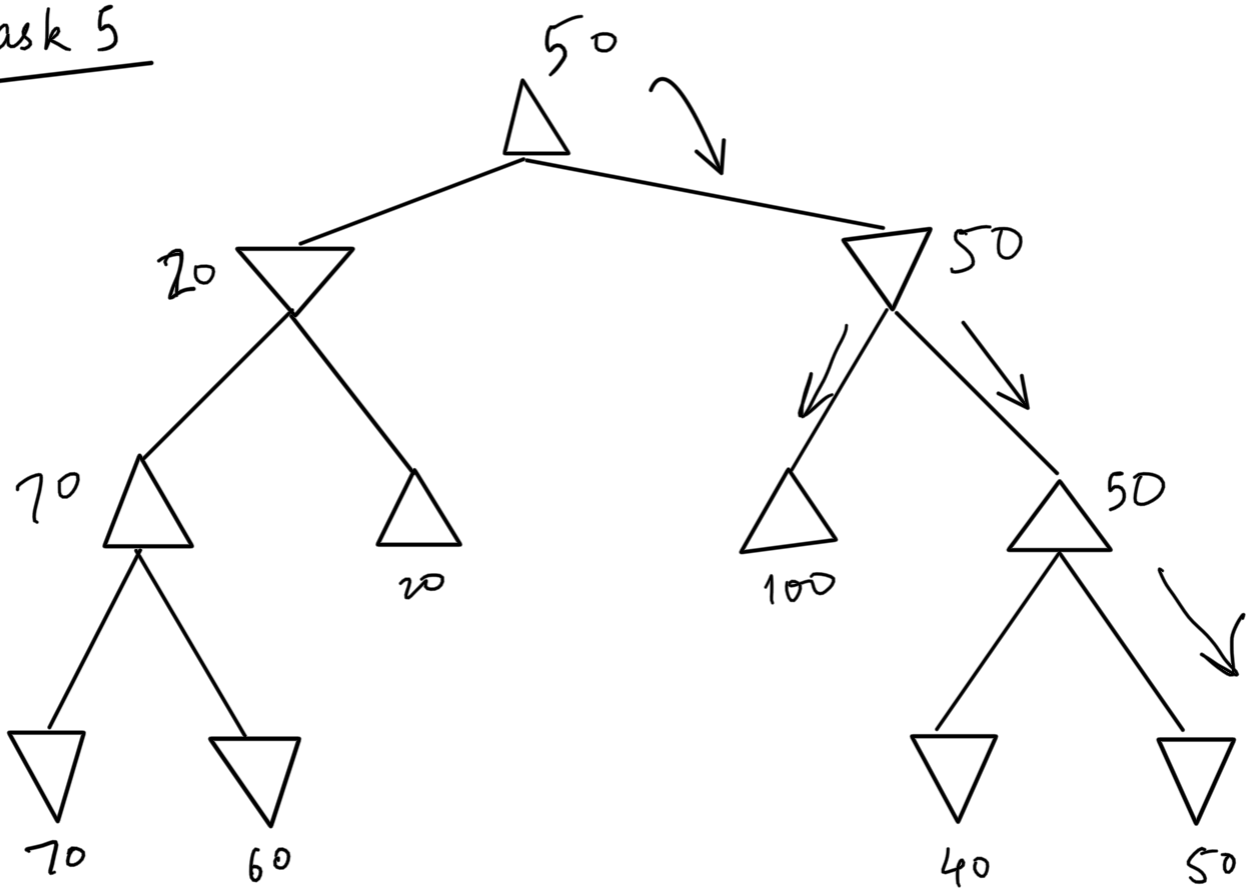




highest = 1000
suboptimal opponent

lowest = 175
taking minimax action

Task 5



In a complete game under the assumptions stated in the question, the worst outcome will be 50 if the opponent plays optimal

strategy.
The best case outcome will be 100 if the opponent plays with a suboptimal strategy for the MAX player.

Task-6

```
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) KB entails S1 if and only if when S1 is true in all words where KB is true.

b) No, it is not necessary for all rows

where KB is false to have S1 false too. There are more than 1 rows in the table where KB is false and S1 is not. So NOT(KB) does not entail NOT(S1).

Task-8

Statement when knowledge base is false.

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

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

Using De'Morgan's theorem

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

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

Task-9

$A \Rightarrow B$ $\quad$ $D \Rightarrow A$ $\quad$ $E \Rightarrow F$

$B \Rightarrow C$ $\quad$ $E \Rightarrow D$ $\quad$ E

$C \Rightarrow B$ $\quad$ $C \Rightarrow F$

i) Forward Chaining

E

D

A

B

C

F

$\frac{E \quad E \Rightarrow D}{D}$ $\frac{D \quad D \Rightarrow A}{A}$

$\frac{A \quad A \Rightarrow B}{B}$ $\frac{B \quad B \Rightarrow C}{C}$

$\frac{C \quad C \Rightarrow F}{F}$

Therefore $KB \not\models F$

ii) Backward Chaining

~~E~~

~~D~~

~~A~~

~~B~~

~~C~~

F

$\frac{E \quad E \Rightarrow D}{D}$ $\frac{D \quad D \Rightarrow A}{A}$

$\frac{A \quad A \Rightarrow B}{B}$ $\frac{B \quad B \Rightarrow C}{C}$

$\frac{C \quad C \Rightarrow F}{F}$

$KB \not\models F$

iii) Resolution

$A \Rightarrow B \quad \neg A \vee B$

$B \Rightarrow C \quad \neg B \vee C$

$C \Rightarrow C \quad \neg C \vee B$

$D \Rightarrow A \quad \neg D \vee A$

$E \Rightarrow D \quad \neg E \vee D$

$C \Rightarrow F \quad \neg C \vee F$

$C \Rightarrow F$ $\quad\quad$ $\neg C \vee F$

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

Add (F) $\neg C$ $\neg B$ $\neg A$ $\neg D$ $\neg E$ □

empty clause

Therefore $KB \not\models F$

Task - 10

a) A : It rains in May
B : John gave Mary a check for $10,000
C : Mary mowed the lawn.

If it rains, it IMPLIES that John gave Mary a check for $10K.

$A \Rightarrow B$

If John gives Mary 10K, it implies that Mary mowed the lawn.

$B \Rightarrow C$

b) NOT $A \wedge B \wedge C$

c) Symbols —

1. A (May)
2. B (John, Mary)
3. B (Mary, John)
4. B (Mary, Mary)
5. B (John, John)
6. C (Mary)

d) $\neg A(May) \wedge B(John, Mary) \wedge C(Mary)$

e) In the above example KB is true and α is also true.

So $KB \models \alpha$ which means contract not violated.