Page No.:
Date: / /

Homework-3

(1)
(a) - Alpha beta pruning is a modified version of minimax algorithm. It is an optimatization technique for the min-max algorithm.

- The min-max algorithm is a recursive algorithm which is used to choose an optimal move.
- When we apply alpha beta pruning to a standard min-max algorithm it returns the same move as the standard one but it removes all the nodes which do not influence the final ~~decis~~ decision.
- Alpha-beta pruning can be applied at any depth of a tree, and sometimes it only not prune the tree leaves but also entire subtree.
- The two parameters can be defined as:

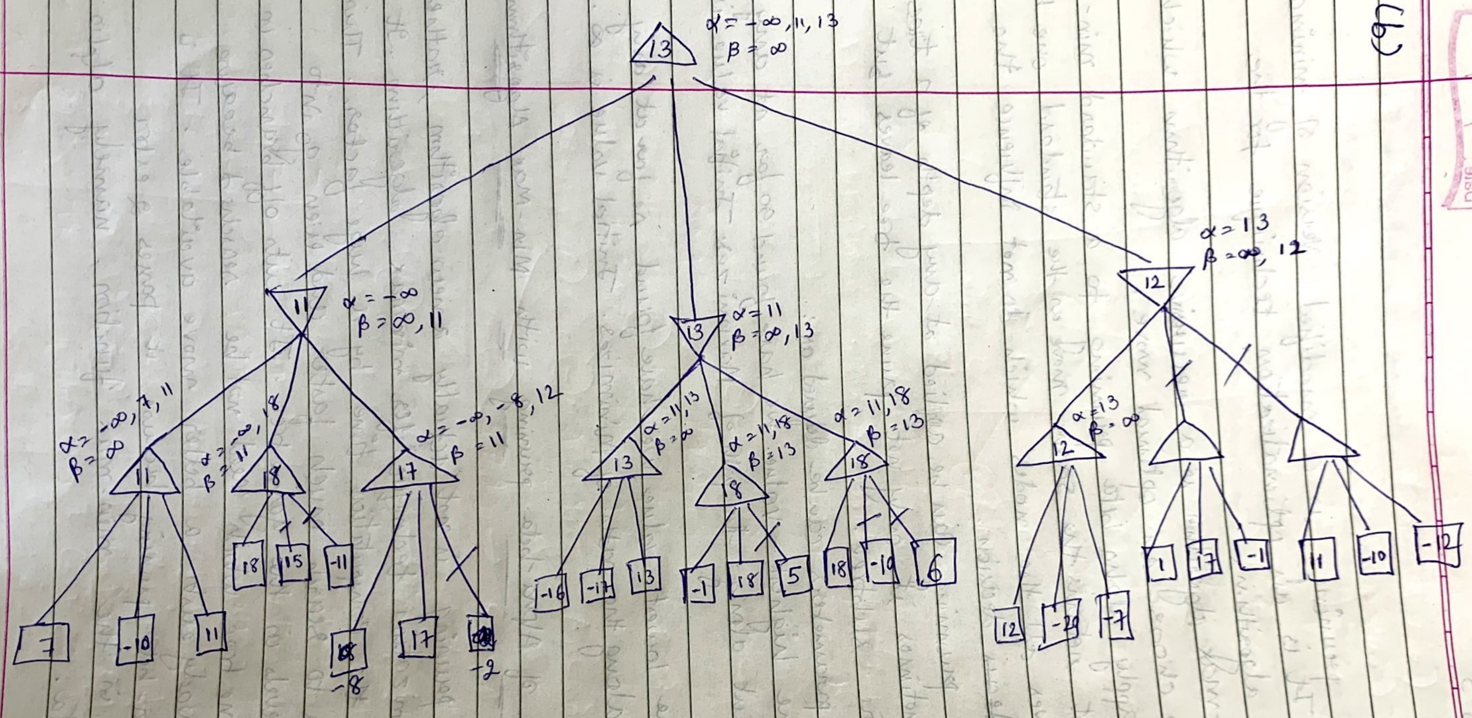
Alpha: The highest-value we have found so far at any point along the path of Maximizer. Initial value is $-\infty$.

Beta: The lowest-value we have found so far at any point along the path of Minimizer. Initial value is ∞.

Comparison of Alpha-beta pruning with Min-Max Algorithm

Alpha beta pruning is not actually a new algorithm, rather an optimization technique for minmax algorithm. It reduces the computation time by a huge factor. This allows us to search much faster and even go into deeper levels in the game tree. It cuts off branches in the game tree which need not be searched because there already exists a better move available. It is called Alpha-beta pruning because it passes 2 extra parameters in the min-max function, namely alpha and beta.

(9)



13
α = -∞, 11, 13
β = ∞
11
α = -∞
β = ∞, 11
13
α = 11
β = ∞, 13
12
α = 13
β = ∞, 12
α = -∞, 7, 11
β = ∞
11
α = -∞, 18
β = 11
18
α = -∞, -8, 12
β = 11
17
13
α = 11, 13
β = ∞
18
α = 11, 18
β = 13
18
α = 11, 18
β = 13
12
α = 13
β = ∞
7
-10
11
18
15
-11
-8
17
-2
-16
-17
13
-1
18
5
18
-19
6
12
-20
-7
1
17
-1
11
-10
-12

(2)

(a) Variables : C_1, C_2, C_3, C_4, C_5 (classes)

Domains : A, B, C

Constraints : Each professor can teach one class at a time

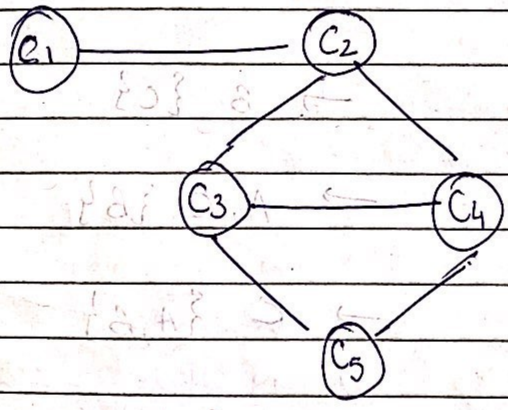
C_1 = 8-9am
C_2 = 8:30-9:30 am
C_3 = 9:00-10:00 am
C_4 = 9:00-10:00 am
C_5 = 9:30-10:30am

$\{C_1 \neq C_2,\ C_2 \neq C_3,\ C_2 \neq C_4,\ C_4 \neq C_5,\ C_3 \neq C_4,\ C_3 \neq C_5\}$

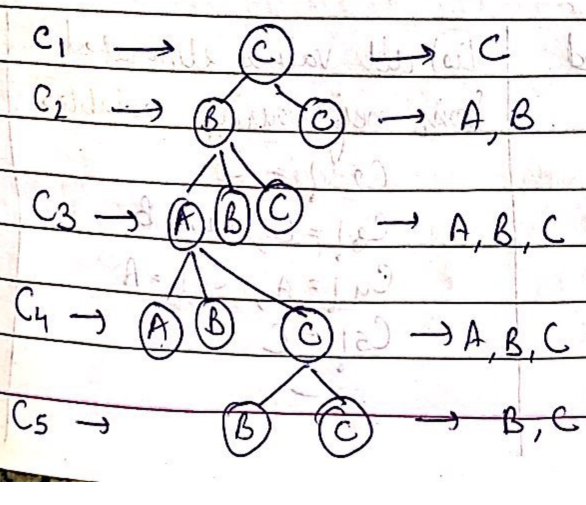
Professor : $C_1 = \{C\}$
$C_2 = \{B, C\}$
$C_3 = \{A, B, C\}$
$C_4 = \{A, B, C\}$
$C_5 = \{B, C\}$

$P_A = C_3, C_4$
$P_B = C_2, C_3, C_4, C_5$
$P_C = C_1, C_2, C_3, C_4, C_5$

(b) Constraint Graph



(c) Backtracking



Page No.:
Date: / /

Step	Var Assigned	list all values eliminated from neighbouring variables	Backtrack
1	$C_1 = C$	None	No
2	$C_2 = B$	None	No
3	$C_3 = A$	None	No
4	$C_4 = A$	None	Yes
5	$C_5 = B$	None	Yes
6	$C_4 = C$	None	No
7	$C_5 = B$	None	No

(d) <u>Backtracking with Forward Chaining</u>

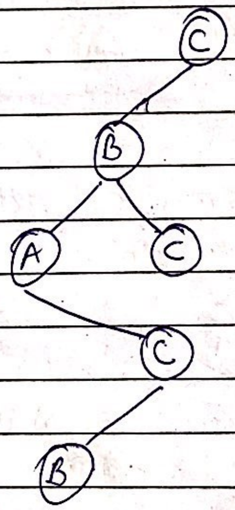
C_1 → C

C_2 → B {C}

C_3 → A, C. {B}

C_4 → C {A, B}

C_5 → B {C}



Step	Var Assigned	list all values eliminated from neighbouring variables	Backtrack
1	$C_1 = C$	$C_2 \neq C$	No
2	$C_2 = B$	$C_3 \neq B$, $C_4 \neq B$	No
3	$C_3 = A$	$C_4 \neq A$, $C_5 \neq A$	No
4	$C_4 = C$	$C_5 \neq C$	No
5	$C_5 = B$	—	No

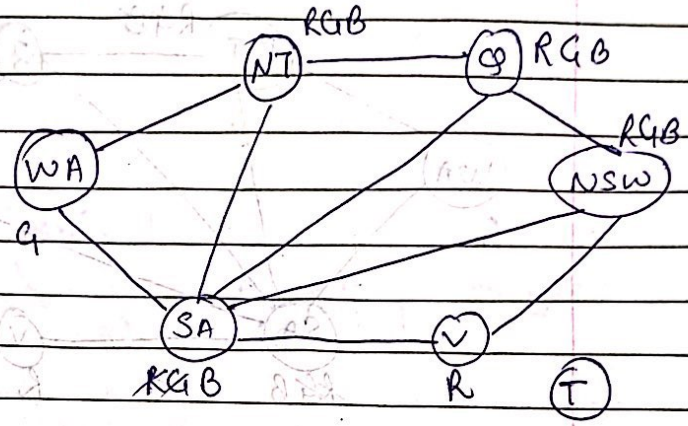
Scanned with CamScanner

(3) Using AC-3 algorithm

Partial assignment {WA = Green, V - Red}

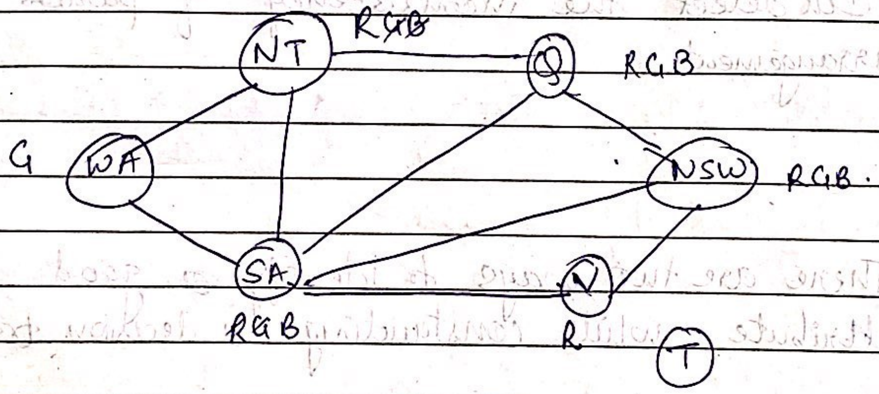
Deleting R from SA

Deleting G from SA.



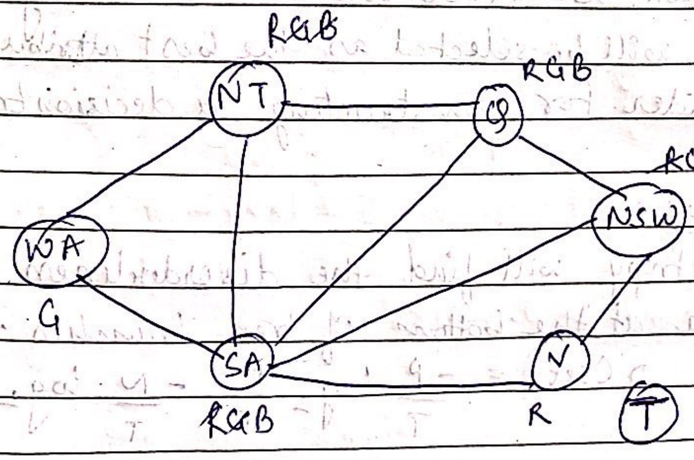
Deleting G from NT (edge WA - NT)

Deleting B from NT (edge SA - NT)

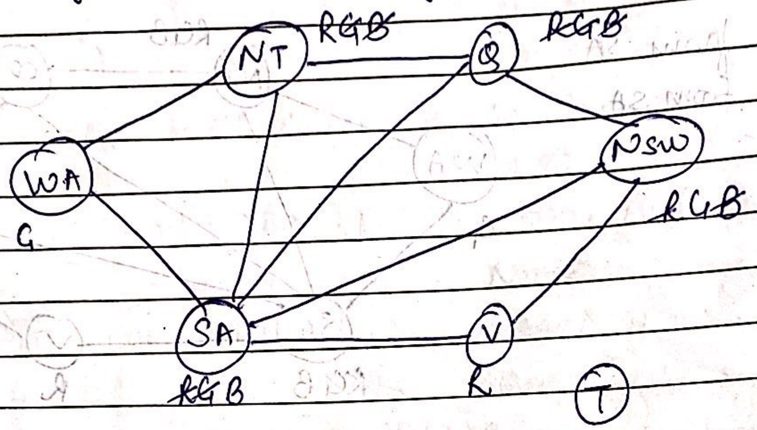


Deleting R from NSW (edge NSW - V)

Deleting B from NSW (edge SA - NSW)



Deleting G from Q (edge Q-NSW)
Deleting B from Q (edge Q-SA)
Deleting R from Q (edge Q-NT)



Queensland (Q) does not have any colors list. So we can conclude that arc consistency can detect the inconsistency of partial arrangement.

(4)

(a) There are two ways to identify a good attribute while constructing a decision tree.

★ Information Gain :
Each attribute splits the nodes into homogeneous children, whichever attribute has a highest score will be selected as the best attribute to consider for constructing a decision tree.

★ Entropy :
Entropy will find the disorderliness of the data at the bottom of tree branches.

$$\therefore D(set) = -\frac{P}{T} \log_2 \frac{P}{T} - \frac{N}{T} \log_2 \frac{N}{T}.$$

here,

P = Positive Value T = Total no. of samples of the
N = Negative Value set of binary values.

$$Q(\text{Test}) = \sum D(\text{set}) \times \left(\frac{\text{No. of samples in set}}{\text{No. of samples handled by test}}\right)$$

(b) Sample (S) = Playing Tennis

Yes (9) No (4)

$$E(\text{playing tennis}) = \sum_{i=1}^{c} -P_i \log_2 P_i$$
$$= 0.94$$

(i) Outlook :- Sunny Overcast Rain

Sunny: Yes (9) No (4) Overcast: Yes (4) No (0) Rain: Yes (3) No (2)

$$E(S_{sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$E(S_{overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$E(S_{rain}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$\text{Gain}(S, \text{outlook}) = E(S) - \frac{5}{14}E(S_{sunny}) - \frac{4}{14}E(S_{overcast}) - \frac{5}{14}E(S_{rain})$$

$$= 0.94 - \frac{5}{14}(0.971) - \frac{4}{14}(0) - \frac{5}{14}(0.971)$$

$$G(S, \text{outlook}) = 0.2464$$

(ii) <u>Temperature</u>: Hot → Yes (2), No (2); Mild → Yes (4), No (2); Cool → Yes (3), No (1)

$E(S_{Hot}) = 1.0$
$E(S_{mild}) = 0.9183$
$E(S_{cool}) = 0.8113$

$G(S, Temp) = 0.0289$

(iii) Humidity: High → Yes (3), No (4); Normal → Yes (6), No (1)

$E(S_{high}) = 0.9852$
$E(S_{normal}) = 0.5916$

$$G(S, Humidity) = 0.94 - \frac{7}{14}(0.9852) - \frac{7}{14}(0.5916)$$

$G(S, Humidity) = 0.1516$

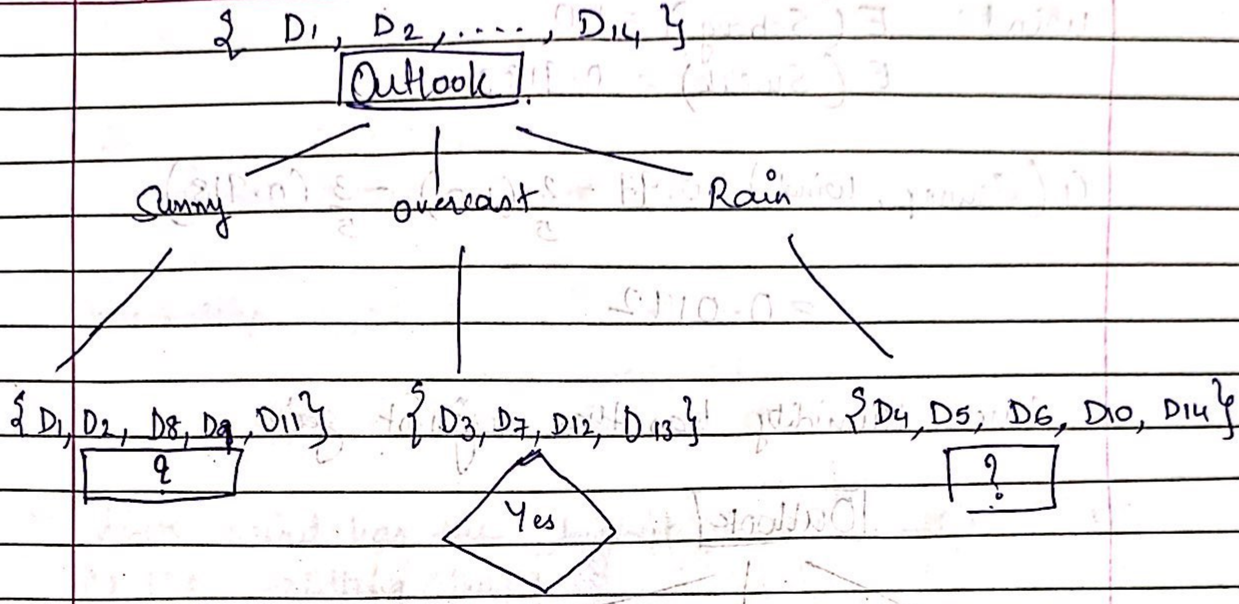
(iv) <u>Wind</u>: Strong → Yes (3), No (3); Weak → Yes (6), No (2)

$E(S_{strong}) = 1.0$
$E(S_{weak}) = 0.8113$

$$G(S, wind) = 0.94 - \frac{6}{14}(1.0) - \frac{8}{14}(0.8113)$$

$G(S, wind) = 0.0478$

<u>Outlook has the highest gain so it is selected for first split as root node.</u>



Splitting attributes = Temperature, Humidity, wind

$E(S_{sunny}) = 0.97$

Temperature : $E(S_{Hot}) = 0.0$
$E(S_{mild}) = 1.0$
$E(S_{cool}) = 0.0$

$$G(S_{sunny}, \text{Temperature}) = 0.97 - \frac{2}{5}(0.0) - \frac{2}{5}(1) - \frac{1}{5}(0.0)$$

$$= \underline{0.570}$$

Humidity : $E(S_{high}) = 0.0$
$E(S_{normal}) = 0.0$

$$G(S_{sunny}, \text{Humidity}) = 0.97 - \frac{3}{5}(0.0) - \frac{2}{5}(0.0)$$

$$= \underline{0.97}$$

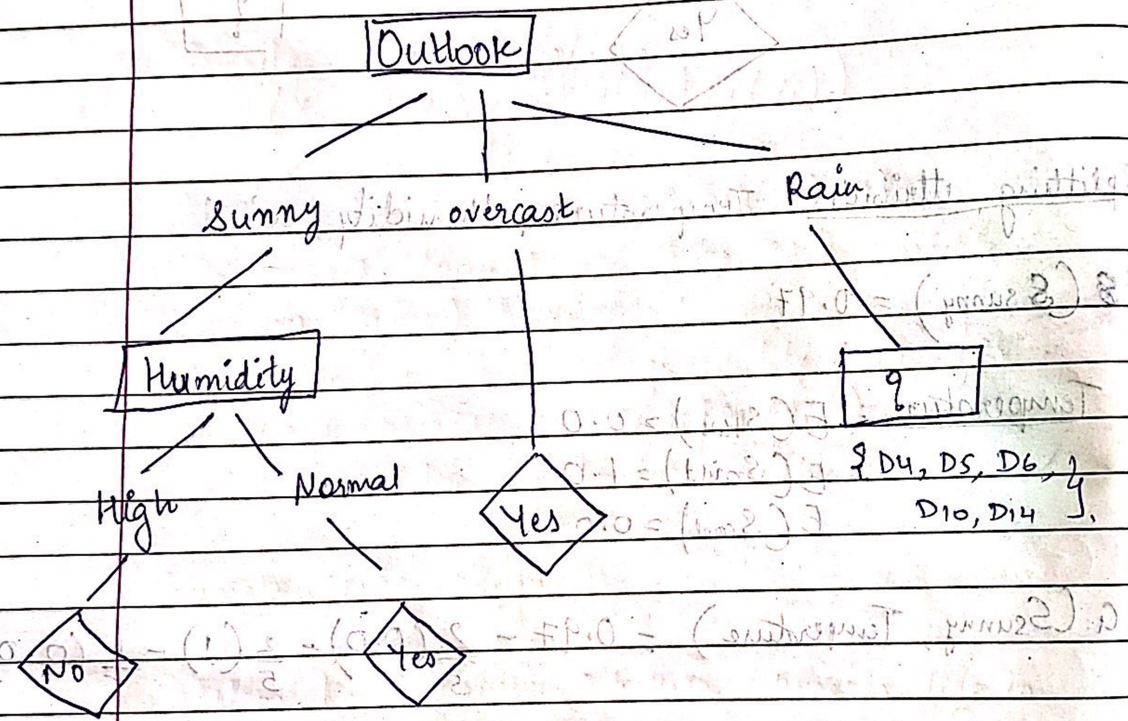
Wind : $E(S_{strong}) = 1.0$

$E(S_{weak}) = 0.9183$

$G(S_{sunny}, \text{wind}) = 0.97 - \frac{2}{5}(1.0) - \frac{3}{5}(0.918)$

$= 0.0192$

Since, Humidity has the highest gain



$E(S_{Rain}) = 0.97$

Temperature : $E(S_{hot}) = 0.0$

$E(S_{mild}) = 0.9183$

$E(S_{cool}) = 1.0$

$G(S_{Rain}, \text{Temperature}) = 0.0192$

Humidity: $E(S_{High}) = 1.0$

$E(S_{normal}) = 0.9183$

$E(S_{cool}) = \cancel{1.0}\ 0.0192$

~~$G(S_{Rain}, Temperature) = 0.0192$~~

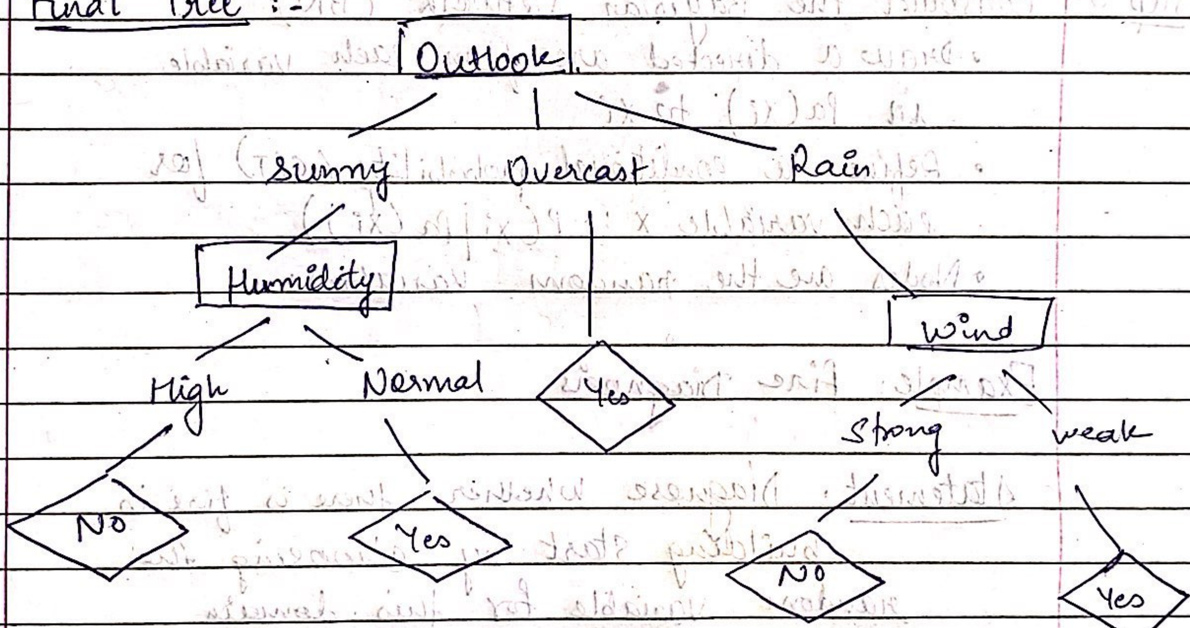
~~Humidity~~ Wind: $E(S_{strong}) = 0.0$

$E(S_{weak}) = 0.0$

$G(S_{Rain}, wind) = 0.97$

Since wind has the highest gain, it will be used as the splitting factor.

Final Tree :-



(5) Bayesian Networks:

Construction:

Building or constructing a Bayesian Network.

Step 1: Define a total order of random variables: $(x_1, x_2, \ldots, x_n)$

Step 2: Apply the chain rule

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_1, \ldots, x_{i-1})$$

Predecessor of x_i in the total order defined over the variables.

Step 3: For each x_i select smallest set of predecessor $Pa(x_i)$ such that
$$P(x_i / X_1, \dots, x_{i-1}) = P(x_i \mid Pa(x_i))$$
x_i is conditionally independent from all its other predecessors given $Pa(x_i)$

Step 4: Rewrite $P(X_1, \dots, X_n)$
$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i))$$

Step 5: Construct the Bayesian Network (BN)
- Draw a directed arc from each variable in $Pa(X_i)$ to X_i
- Define a conditional probability (CPT) for each variable X: $P(X_i \mid Pa(X_i))$
- Nodes are the random variables.

Example: Fire Diagnosis

Statement: Diagnose whether there is fire in building start by choosing the random variable for this domain

- Tampering (T), Fire (F), Alarm (A), Smoke (S), Leaving (L), Report (R).

Step 1: F, T, A, S, L, R

Step 2: $P(F,T,A,S,L,R) = P(F)\,P(T/F)\,P(A/F,T)\,P(S \mid F,T,A)\,P(L / F,T,A,S)\,P(R \mid F,T,A,S,L)$

Step 3: $P(X_i \mid X_1, \ldots, X_{i-1}) = P(X_i \mid Parents(X_i))$

Step 4: $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Parents(X_i))$

Step 5: Bayesian Network