**Discrete Mathematical Structures**

**Week-15**

**Long Descriptive Questions**

**1. State and Prove De Morgan's laws of Boolean algebra. Also Simplify the Boolean expression [((x.y').(z+y.w)+ x'.y').z].**

De Morgan's laws are a set of two fundamental principles in Boolean algebra that relate to the complement (negation) of logical expressions. These laws are named after the mathematician Augustus De Morgan and are widely used in logic and digital circuit design. The two laws are as follows:

**De Morgan's First Law**

* The complement of the union (OR) of two sets is equal to the intersection (AND) of their complements.
* Symbolically, for any two logical expressions A and B, De Morgan's First Law can be stated as̴(A Ս B) = ( ̴A) Ո ( ̴B)

**De Morgan's Second Law**

* The complement of the intersection (AND) of two sets is equal to the union (OR) of their

Complements.

* Symbolically, for any two logical expressions A and B, De Morgan's Second Law can be stated as ̴(A Ո B) = ( ̴A) Ս ( ̴B)

Now, let's prove De Morgan's First Law

Proof of De Morgan's First Law

To prove ̴(A Ս B) = ( ̴A) Ո ( ̴B), we need to show that both sides are equal using logical equivalences.

1. Start with the left-hand side (LHS): ̴(A Ս B)
2. Apply the negation ( ̴) to the union (A Ս B): ̴ (A Ս B)= ̴A Ո ̴B
3. The result on the right-hand side (RHS) is ( ̴A) Ո ( ̴B), which matches the desired expression.

Therefore, we have proved De Morgan's First Law: ̴(A Ս B) = ( ̴A) Ո ( ̴B).

Next, let's simplify the Boolean expression [((x.y').(z+y.w) + x'.y').z]:

[((x.y').(z+y.w) + x'.y').z]

= [((x.y').z + (x.y').(y.w)) + (x'.y').z]

= [(x.y'.z + x.y'.y.w) + (x'.y').z]

= [(x.y'.z + x.O.w) + (x'.y').z]

= [(x.y'.z + 0) + (x'.y').z]

= (x.y'.z + x'.y'.z)

Now, let's distribute and simplify further:

(x.y'.z + x'.y.z)

= [(x.y' + x'.y') . z]

= [(x.y' + x'.y') . z]

= [(x + x').(y' + y') . z]

= [1 .1 . z]

=z

Therefore, the simplified Boolean expression [((x.y').(z+y.w) + x'.y'.z] simplifies to z.