Group:

Muhammad Zaki Mufthi

Vico King

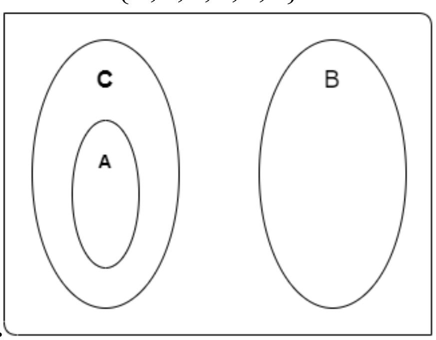
Rizal Rafiuddin

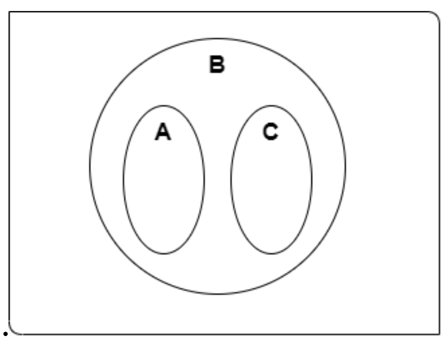
1. A={1,2}
   B={1,2,3}
   C={3,4,5,6,7,8}

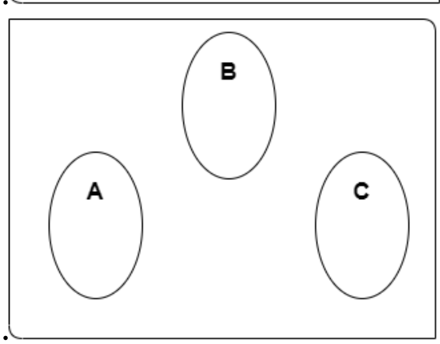
   a.A∪C={1,2,3,4,5,6,7,8}

   b.(A∪B)'={4,5,6,7,8}

   c.A'∪B'={3,4,5,6,7,8}

2. A. 

   B. 

   C. 

3. A={-1,1,2,4}
   B={1,2}
   AxB={(-1,1),(-1,2),(1,1),(1,2),(2,1),(2,2),(4,1),(4,2)
   S={(-1,1),(1,1),(2,2)}
   T={(-1,1),(1,1),(2,2),(4,2)}
   S∪T={(-1,1),(1,1),(2,2),(4,2)}
   S∩T={(-1,1),(1,1),(2,2)}

4. $P \equiv \neg\left((\neg p \wedge q) \vee (\neg p \wedge \neg q)\right) \vee (p \wedge q)$

   $P \equiv \neg(\neg p \wedge q) \wedge \neg(\neg p \wedge \neg q) \vee (p \wedge q)$ (De Morgan's Law)

   $P \equiv (\neg\neg p \vee \neg q) \wedge \neg\neg(p \vee q) \vee (p \wedge q)$ (Double Negation Law)

   $P \equiv (p \vee \neg q) \wedge (p \vee q) \vee (p \wedge q)$ (Distributive Law)

   $P \equiv p \vee (\neg q \wedge q) \vee (p \wedge q)$ (Complement Law)

   $P \equiv p \vee \emptyset \vee (p \wedge q)$ (Property of Empty Set)

   $P \equiv p \vee (p \wedge q)$ (Absorption Law)

   $P \equiv P$

5. A)

$R1$	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	0
3	1	1	1	0	0
4	1	1	0	0	0
5	1	0	0	0	0

B)
$$\begin{array}{cccccc} R2 & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 1 & 0 & 0 \\ 5 & 1 & 1 & 1 & 1 & 0 \end{array}$$

C) R1= not reflexive, symmetric, transitive, and not equivalence relation

D) R2= not reflexive, antisymmetric, not transitive, and not partial order relation

6. A) $\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}$ B) $\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$

7. Let: x1,x2∈R be two distinct elements

then g(x1)≠g(x2), as g is one-to-one function.

And **f**(**g**(**x1**))≠**f**(**g**(**x2**)) as **f** is also one-one function.

f+**g** may not one-to-one functions even if **f** and **g** are one-to-one.

consider **f**(**x**)=**x** and **g**(**x**)= −**x**, then **f**+**g** then is not one-to-one.

8.

$C_1=1$

$C_2=2$

$n\geq 1$

the recurrence:

C_n=Cn-1+Cn-2.

9.

a). $T_{7=}$ 13

b). Recursive algorithm:

Input n

Output t(n)

```
T(n){
If (n=0)
        return 0
Else if (n=1 or n=2)
        return 1
return t (tn=tn-1+tn-2+tn-3 )
```