UTM
UNIVERSITI TEKNOLOGI MALAYSIA
SCHOOL OF COMPUTING
Faculty of Engineering

ASSIGNMENT 1

DISCRETE STRUCTURE-07

GROUP-01:
Muhammad Aqila Karindra Daffa (A19EC3010)
Zhou Chenxing (A19EC4034)
Muhammad Najib bin Jamaludin (A20EC0215)

Q_1

a) $A \cup C = \cancel{\emptyset}\{x \in \mathbb{R} \mid 0 < x < 9\}$

b) $A \cup B = \{x \in \mathbb{R} \mid 0 < x < 4\}$

$(A \cup B)' = \{x \in \mathbb{R} \mid x \leq 0, x \geq 4\}$

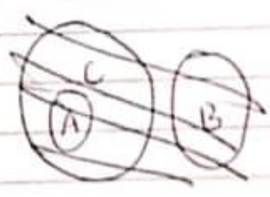
c) $A' = \{x \in \mathbb{R} \mid x \leq 0, x > 2\}$

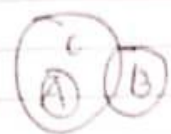
$B' = \{x \in \mathbb{R} \mid x < 1, x \geq 4\}$

$A' \cup B' = \{x \in \mathbb{R} \mid x < 1, x > 2\}$

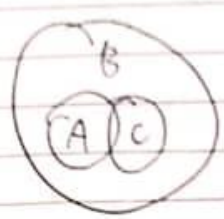
Q_2

a) $A \cap B = \emptyset$, $A \subseteq C$, $C \cap B = \emptyset$

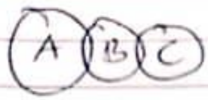




b) $A \subseteq B$, $C \subseteq B$, $A \cap C \neq \emptyset$



c) $A \cap B \neq \emptyset$, $B \cap C \neq \emptyset$, $A \cap C = \emptyset$, $A \not\subset B$, $C \not\subset B$



$\cancel{\neg((\neg p \wedge q) \vee (\neg p \wedge \neg q)) \vee (p \wedge q) \equiv p}$

Q_3

$A = \{-1, 1, 2, 4\}$ $B = \{1, 2\}$

1) $A \times B$

$= \{(-1,1), (-1,2), (1,1), (1,2), (2,1), (2,2), (4,1), (4,2)\}$

2) S

$|x| = |y|$

$= \{(-1,1), (1,1), (2,2)\}$

3) T

$x-y$ is even $= \{(-1,1), (1,1), (2,2), (4,2)\}$

4) $S \cap T = \{(-1,1), (1,1), (2,2)\}$

5) $S \cup T = \{(-1,1), (1,1), (2,2), (4,2)\}$

Q4) Show that $\neg((\neg p \wedge q) \vee (\neg p \wedge \neg q)) \vee (p \wedge q) \equiv p$

* $\neg((\neg p \wedge q) \vee (\neg p \wedge \neg q)) \vee (p \wedge q) \equiv p$

* $\neg(\neg p \wedge (q \vee \neg q)) \vee (p \wedge q) \equiv p$

↳ // Distributive laws : $[(p \wedge q) \vee (p \wedge r) \equiv p \wedge (q \vee r)]$

* $p \vee \neg(q \vee \neg q) \vee (p \wedge q) \equiv p$

↳ // De Morgan's Law : $[\neg(p \wedge q) \equiv (\neg p) \vee (\neg q)]$

* $p \vee (p \wedge q) \equiv p$

↳ // Logic & Sets : $[p \wedge \neg(q \wedge \neg q) \equiv p]$

* $p \equiv p$ //

↳ // Absorption Laws : $[p \vee (p \wedge q) \equiv p]$

Thus, $\neg((\neg p \wedge q) \vee (\neg p \wedge \neg q)) \vee (p \wedge q) \equiv p$

Question 5

a) $\underset{X \text{ to } Y}{R_1} = \{ \cancel{(x+y)}\ (x,y) \mid x+y \leq 6 \}$

$X = \{1, 2, 3, 4, 5\}$
$Y = \{1, 2, 3, 4, 5\}$

$$A_1 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix}$$

b) $\underset{Y \text{ to } Z}{R_2} = \{ (y, z) \mid y > z \}$

$Z = \{1, 2, 3, 4, 5\}$

$$A_2 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix}$$

c) A_1 = Transitive
d) A_2 = Transitive

Question 6

a) $R_1 \cup R_2 = \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}$

b) $R_1 \cap R_2 = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix}$

Question 7

f+g might not be one ~~tor~~ one.

consider,

$f(x) = x \quad , \quad g(x) = -x$

$(f+g)(x) = f(x) + g(x)$
$= x + (-x)$
$= 0$

let $(f+g)(\cancel{x}\ 2) = f(2) + g(2)$
$= 2 + (-2)$
$= 0$

We can see that $(f+g)(x) = 0$; for all real number of x.

$\therefore$ f + g is not one to one.

Scanned with CamScanner

Q8) n = The Stairs

C_n = The number of different ways to claimb the stairs

Let $C_1 = 1$, $C_2 = 2$

if C_n = The number of different ways to claimb the stairs then n:

$C_1, C_2, C_3, C_4, C_5, C_6$

$1, 2, 3, 5, 8, 13$

because it can be 1 stair at a time, 2 stairs at a time, or taking a combination of one or two strair increments.

So,: Since $C_1 - C_6$:

$$\underbrace{1, 2}_{1}, \overset{2+1}{3}, \overset{3+2}{5}, \overset{5+3}{8}, \overset{8+5}{13} \Rightarrow \boxed{C_n = C_{n-1} + C_{n-2} \, , \; n \geq 3}$$

1 1 2 3 5

Q 9) The tribonacci sequence (t_n) is defined by equation

$t_0 = 0, \; t_1 = t_2 = 1, \; t_n = t_{n-1} + t_{n-2} + t_{n-3}, \; n \geq 3$

a. t_7 ?

$t_3 = t_2 + t_1 + t_0$

$= 1 + 1 + 0 = 2$

$t_4 = t_3 + t_2 + t_1$

$= 2 + 1 + 1 = 4$

$t_5 = t_4 + t_3 + t_2$

$= 4 + 2 + 1 = 7$

$t_6 = t_5 + t_4 + t_3$

$= 7 + 4 + 2 = 13$

$t_7 = t_6 + t_5 + t_4$

$= 13 + 7 + 4 = 20$

b. * Input = n

* Output = t_n

* t(n) {

```
if ( n = 0 ){
    return 0
    return t(n)=t(n-1) + t(n-2) + t(n-3) }
```

* if (n = 1 and n = 2){

```
    return 1
    return t(n) = t(n-1) + t(n-2) + t(n-3) }}
```