Name: Archit Agrawal
Student ID: 202052307

Date:
Page: 1

MA102: Introduction to Discrete Mathematics
Tutorial 3

1. Which of these sentences are propositions? What are the truth value of those that are propositions?

(a) Answer this question.
(b) This statement is true.
(c) $6 + 5 = 11$
(d) What time is it?
(e) Close the door
(f) $x = x$

(a) not a proposition
(b) not a proposition
(c) proposition (true)
(d) not a proposition
(e) not a proposition
(f) proposition (true)

2. What is the negation of each of these propositions?

(a) Surya and Tejay are friends.
(b) The summer in Gandhinagar is hot and sunny.
(c) Shubham sent more than 100 whatapp messages everyday.
(d) $5 * 3 \geq 15$
(e) It is freezing and it is not snowing.

(a) Surya and Tejay are not friends.

(b) The summer in Gandhinagar is not hot and sunny.

(c) Shubham sent at most 100 whatsapp messages every day.

(d) $5 * 3 < 15$

(e) It is not freezing or it is snowing.

3. Let p and q be the propositions :-

p : Mahesh choose MA401 as science elective.
q : Mahesh likes MA102

Express each of these propositions as English sentence.

$p \vee q$, $\neg p \wedge \neg q$, $p \rightarrow q$

$p \vee q$:⇒ Mahesh choose MA401 as science elective or Mahesh likes MA102.

$\neg p \wedge \neg q$:⇒ Mahesh didn't choose MA401 as science elective and Mahesh does not like MA102.

$p \rightarrow q$:⇒ If Mahesh choose MA401 as science elective, then Mahesh likes MA102.

Name: Archit Agrawal
Student ID: 202052307

4. Write these propositions using p and q and logical connectives.

p: Mahesh choose MA 401 as science elective.
q: Mahesh likes MA 102.

(a) Mahesh does not choose MA 401 and likes MA 102.
(b) Either Mahesh does not like MA 102 or he does not choose MA 401.

(a) $\neg p \wedge q$
(b) $\neg p \vee \neg q$

5. Let p, q, r be three propositions with truth values F, T, F respectively. Find the truth values of

(a) $p \rightarrow \neg r$
(b) $p \vee \neg r$
(c) $(p \wedge \neg q) \rightarrow r$
(d) $(r \rightarrow \neg p) \rightarrow q$

(a) $p \rightarrow \neg r$ is true, as p is false.

(b) $p \vee \neg r$

truth value of p is false
truth value of r is false.
Hence, truth value of $\neg r$ is true.

$\therefore$ truth value of $p \vee \neg r$ is true.

(c) $(p \wedge \neg q) \rightarrow r$

truth value of

As p is false, $(p \wedge \neg q)$ is false

Hence, truth value of $(p \wedge \neg q) \rightarrow r$ is true.

(d) $(r \rightarrow \neg p) \rightarrow q$

As truth value of r is false, the truth value of $(r \rightarrow \neg p)$ is true.

∴ truth value of q is true, the truth value of $(r \rightarrow \neg p) \rightarrow q$ is true.

6. Is f(n) = largest prime factor of n is a computable function from N to N? If yes then write down a C program for it.

Yes, it is a computable function from N to N.

```c
#include <stdio.h>
int largestPrimeFactor(int n){
    for(int i = n; i >= 2; i--){
        if(n % i == 0 && isPrime(i) == 1){
            return i;
        }
    }
    return -1;
}

int isPrime(int x){
    int count = 0;
    for(int p = 2; p < x/2 + 1; p++){
        if(x % p == 0) count++;
    }
    if(count != 0) return 0;
    else return 1;
}

int main(){
    printf("%d", largestPrimeFactor(105));
    return 0;
}

```

Name: Archit Agrawal
Student ID: 202052307

Date:
Page: 5

7. Is $\{n^2 \mid n \in N\}$ a computable set?

A set is called computable (or decidable) if there is an algorithm which takes a number as input, terminates after a finite amount of time and correctly decides whether the number belongs to the set or not.

Let there be a function $f(n)$ whose output is 1 if $n = K^2$ $(K \in N)$ and else its output is zero. i.e.

$$f(n) = \begin{cases} 1 & \text{if } n = K^2 \text{ for } K \in N \\ 0 & \text{if } n \neq K^2 \text{ for } K \in N \end{cases}$$

The below written C function determines if an element belongs to the given set or not.

```
int belongsToSet(int n){
    int i = 1;
    while (i*i <= n){
        if (i*i == n) return 1;
        i++;
    }
    return 0;
}
```

Hence, the given set is computable.

Name: Archit Agrawal
Student ID: 2020052307

Date:
Page: 6

Q. What can you say about cardinality of $\{\mathbb{R}, \mathbb{C}\}$

The cardinality of $\mathbb{R}$ and $\mathbb{C}$ is $\aleph_1$ (aleph 1).

Proof:

Let $f : \mathbb{C} \to \mathbb{R} \times \mathbb{R}$

$f(a+ib) = (a,b)$

The above function is clearly bijective.

Now, let us prove that

$|\mathbb{R} \times \mathbb{R}| = |\mathbb{R}|$

as $|\mathbb{R}| = |[0,1)|$

we can prove that

$|[0,1) \times [0,1)| = |[0,1)|$

Let $f_1 : [0,1) \to [0,1) \times [0,1)$

$f_1(x) = (x, 0)$

Clearly for every x in $[0,1)$, there is unique element in $[0,1) \times [0,1)$, hence f_1 is injective and surjective (ie. bijective)

Now, let $f_2 : [0,1) \times [0,1) \to [0,1)$

$x = 0.a_1a_2a_3\cdots$

$y = 0.b_1b_2b_3\cdots$

$f_2(x,y) = 0.a_1b_1a_2b_2a_3b_3\cdots$

This function is also bijective.

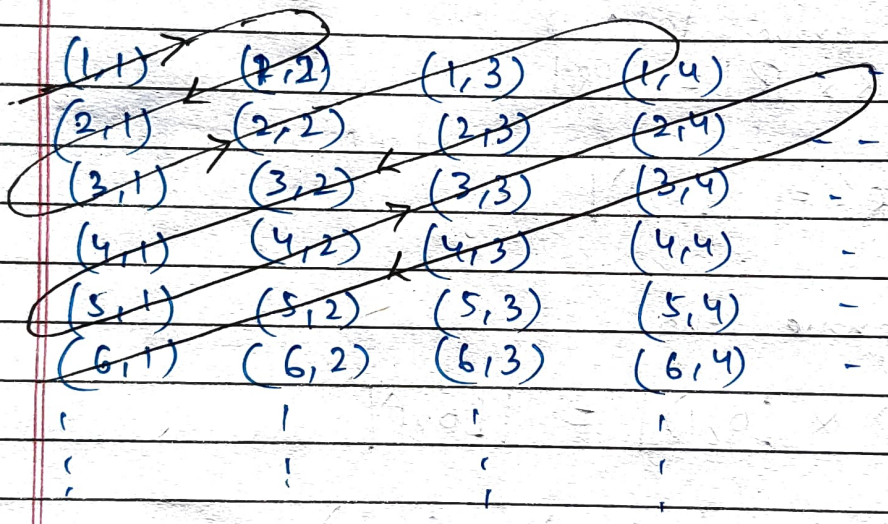
$\therefore |\mathbb{R}| = |\mathbb{R} \times \mathbb{R}|$ — (I)

and $|\mathbb{C}| = |\mathbb{R} \times \mathbb{R}|$ — (II)

from (I) & (II)

$|\mathbb{R}| = |\mathbb{C}| = \aleph_1$

9. Show that $|\mathbb{N} \times \mathbb{N}| = \aleph_0$.



Let $f(n) = n^{th}$ element along the path given above where $n \in \mathbb{N}$

Hence, we have a bijective mapping from $\mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$

$\therefore |\mathbb{N}| = |\mathbb{N} \times \mathbb{N}| = \aleph_0$

Name: Archit Agrawal
Student ID: 202052307

Date:
Page: 8

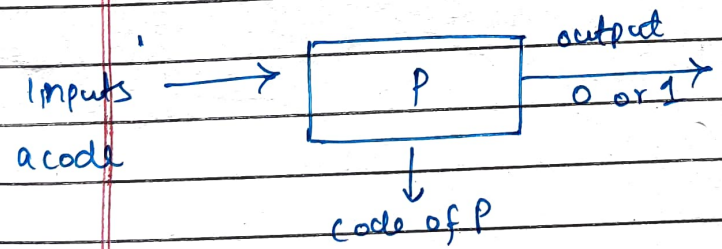
10. Explain the Halting problem. Is it computable?

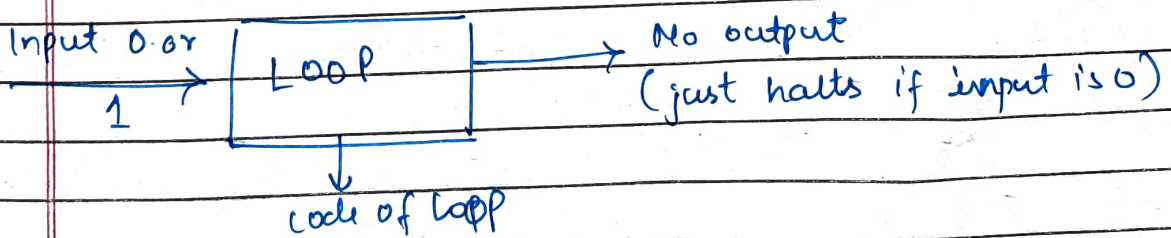
The halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running, or continue to run forever.

Alan Turing proved in 1936 that Halting problem is not computable.

Proof: Let there be a program P, which takes the input as another program and outputs 1 if it halts and 0 otherwise.

Now, let there be another program called LOOP, which loops forever if the input is 1 and stops immediately if input is 0.





Name: Archit Agrawal
Student ID: 20205 2307

Now, let us consider the program, P + LOOP

When we input the code of P+LOOP to itself, and assume that P outputs 1, but after it outputs 1 to LOOP, the LOOP doesn't halt; meaning P should have outputted 0 instead of 1.

∴ We have a contradiction.

∴ A program P which certainly decides whether any given program will halt or not cannot exist.