Discrete Structures (CS 1005)

Group Member 1: Muhammad Faizan Shabir

Roll Number: 22L-6552

Group Member 2:

Roll Number:

Section: 3H

Date: 9th Oct, 2023

Assignment Number: 2 (Two)

Q1) Part 1) Determine whether these system specifications are consistent:

a) "The diagnostic message is stored in the buffer or it is retransmitted."

b) "The diagnostic message is not stored in the buffer."

c) "If the diagnostic message is stored in the buffer, then it is retransmitted."

Let b : "The diagnostic message is stored in the buffer"

Let r : "The diagnostic message is retransmitted"

① The first specification is $b \vee r$ (given)

② The second specification is $\neg b$ (given)

③ The third specification is $b \rightarrow r$ (given)

④ From ② if $\neg b$ is true then b is false

⑤ If ① is true and b is false then r is true (Disjunctive Syllogism)

⑥ Since b is false and r is true then $b \rightarrow r$ must also be true.

Finally, all the three system specifications are true when b is false and r is true i.e.

$\rightarrow b \vee r$ is true for b = false and r = true

$\rightarrow \neg b$ is true for b = false and r = true

$\rightarrow b \rightarrow r$ is true for b = false and r = true

So we can say that the system specifications are consistent.] Answer

Q 1) Part 2) Determine whether these system specifications are consistent:

a) "The diagnostic message is stored in the buffer or it is retransmitted."

b) "The diagnostic message is not stored in the buffer."

c) "If the diagnostic message is stored in the buffer, then it is retransmitted."

d) "The diagnostic message is not retransmitted"

Let b: "The ~~dig~~ diagnostic message is stored in the buffer"

Let r: "The diagnostic message is retransmitted"

① The first specification is $b \vee r$ (given)

② The second specification is $\neg b$ (given)

③ The third specification is $b \rightarrow r$ (given)

④ The fourth specification is $\neg r$ (given)

⑤ from ② if $\neg b$ is true then b is false

⑥ from ④ if $\neg r$ is true then r is false

⑦ $b \rightarrow r$ is true since b is false and r is false

⑧ if ① is true then either b should be true or r should be true but we see that both b and r are false so ① is false.
If we ~~ass~~ assume b to be true then ② becomes false
If we assume r to be true then ④ becomes false.

Finally, we cannot find any combination of truth values for b and r which make all of the specifications true at ~~once~~ once.

b	r	① $b \vee r$	② $\neg b$	③ $b \to r$	④ $\neg r$
F	F	F	T	T	T
F	T	T	T	T	F
T	F	T	F	F	T
T	T	T	F	T	F

So there is not even a single row in the above truth table which is true for all the specifications.

So we can conclude that the system specifications are not consistent.

Answer

Q2) You are given as input a function "int F(int x)" and two arrays, int D[n] containing elements of the domain of the function and int C[m] containing elements of the codomain of the function (Note: n and m could be the same number). Write a code fragment to determine whether or not the function is :

Q2) Part 1) One-to-One

```
bool OneToOne(int D[], int n, int C[], int m)
{
    int image;
    int preImageCount;

    For(int i=0; i<m; i++)
    {
        image = C[i];
        PreImageCount = 0;
        For(int j=0; j<n; j++)
        {
            if(image == F(D[j]))
            {
                PreImageCount++;
            }
            if(preImageCount > 1)
            {
                return false;
            }
        }
    }
    return true;
}
```

Q2) Part 2) Onto

```
bool Onto (int D[], int n, int C[], int m)
{
    int image;

    int PreImageCount;

    for(int i=0; i<m; i++)
    {
        image = C[i];

        PreImageCount = 0;

        for(int j=0; j<n; j++)
        {
            if(image == F(D[j]))
            {
                PreImageCount++;
            }
        }

        if( PreImageCount == 0)
        {
            return false;
        }
    }
    return true;
}
```

Q2) Part 3) One-to-One and Onto (One-to-One Correspondence)

```
bool OneToOneCorrespondence(int D[],
        int n, int C[], int m)
{
        if(OneToOne(D,n,C,m) && Onto(D,n,C,m))
        {
                return true;
        }
        else
        {
                return false;
        }
}
```

Q 3) Show ~~the~~ ~~that~~ that the $|(0,1)| = |(0,1]|$

The Schroder-Bernstein Theorem says: If there are one to one functions f from A to B and g from B to A, then there is one to one correspondence between A and B.

Finding a one-to-one function from $(0,1)$ to $(0,1]$

$f(x) = x$ is a one to one function from $(0,1)$ to $(0,1]$

$f(x) = x$ maps every element of $(0,1)$ to an image that excists in $(0,1]$

Finding a one-to one function from $(0,1]$ to $(0,1)$

$g(x) = \frac{x}{2}$ is a one to one function from $(0,1]$ to $(0,1)$

$g(x) = \frac{x}{2}$ maps every element of $(0,1]$ to an image that excists in $(0,1)$

Since we have found a one to one function $f(x) = x$ from $(0,1)$ to $(0,1]$ and a one to one function $g(x) = \frac{x}{2}$ from $(0,1]$ to $(0,1)$, we can use the Schroder-Bernstein theorem to conclude that there is a one to one correspondence between $(0,1)$ and $(0,1]$

Since we have established a one to one correspondence between $(0,1)$ and $(0,1]$, it automatically implies that $(0,1)$ and $(0,1]$ have the same cardinality

$$\Rightarrow \quad |(0,1)| = |(0,1]|$$

Q4) While studying the countable sets; we saw the use of dovetailing technique to show that the set of positive rational numbers is countable. Write a computer program that prints first 1000 positive rational numbers using the method described in the book.

Note: Your program should not print a number more than one time; e.g. in the figure above, the numbers that are not circled should not be printed by your program.

// assuming that there is a built in function "gcd" that returns greatest common divisor of two numbers

```
int main()
{
    int DiagonalNum = 1;
    int Counter;
    int  P;
    int  q;

    for(Counter=0; Counter< 1000;)
    {
        if( DiagonalNum % 2 != 0) // number is odd
        {
            P= Diagonal Num;
            q = 1;

            do
            {
                if(gcd(P,q)==1)
                {
                    Counter++;
                    Cout<< Counter << ":" <<
                    P<< "/" << q << endl;
                }

                --P; // decrement P
                ++q; // increment q
            }while( P>0 && q < DiagonalNum+1
                    && Counter<1000)
```

```
        ++DiagonalNum; //move to next diagonal
    } // end of IF block
    else // Number is even
    {
        P = 1;
        q = DiagonalNum;

        do
        {
            if(gcd(P, q) == 1)
            {
                Counter++;
                Cout << Counter << ":" <<
                P << "/" << q << endl;
            }
            ++P;   // increment P
            --q;   // decrement q
        } while(P < DiagonalNum + 1 &&
                q > 0 && Counter < 1000)

        ++DiagonalNum; //move to next diagonal
    } // end of ELSE block
} // end of for loop
return 0;
} // end of main
```