Discrete Structures (CS 1005)

Name: Muhammad Faizan Shabir

Roll Number: 22L-6552

Section: 3H

Date: 23rd October, 2023

Assignment Number: 3 (Three)

Q 1) You are given as input a 0-1 matrix M[n][n] representing a relation R on a set. Write a code fragment to determine whether or not the relation R is Transitive.

```
bool IsTransitive(int M[n][n])
{
    int MSquared[n][n];
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            MSquared[i][j] = 0;
            for(int K=0; K<n; K++)
            {
                MSquared[i][j] +=
                M[i][K] * M[K][j];
            }
            if(MSquared[i][j] != 0)
            {
                MSquared[i][j] = 1;
            }
        }
    }
}
```

```
for(int i=0; i< n; i++)
{
    for(int j=0; j<n; j++)
    {
        if(MSquared[i][j]==1 && M[i][j]==0)
        {
            return false;
        }
    }
}
return true;
}
```

(Q2) You are given as input a 0-1 matrix M[n][n] representing a relation R on a set. Write a code fragment to compute the Transitive Closure of R.

```
Void ComputeTransitiveClosure(int M[n][n],
                              int TransitiveClosure[n][n])
{
    int tempMatrix[n][n]; // temporary matrix
    int powerMatrix[n][n];//matrix to hold matrix powers

    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            powerMatrix[i][j] = M[i][j];
            // initialize power matrix with M[n][n]

            TransitiveClosure[i][j] = M[i][j];
            //initialize transitive closure matrix
              with M[n][n]
        }
    }
}
```

```
// loop runs (n-1) times
for(int counter=1; counter<n; counter++)
{
    // loop to calculate powers of matrix
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            // reset temporary matrix entry to 0
            tempMatrix[i][j] = 0;

            for(int K=0; K<n; K++)
            {
                tempMatrix[i][j] +=
                powerMatrix[i][K] * M[K][j];
                // calculation of one entry of
                the power matrix
            }

            if(tempMatrix[i][j] != 0)
            {
                tempMatrix[i][j] = 1;
            }

        }

    }
```

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        powerMatrix[i][j]=tempMatrix[i][j];
        // Copy the calculated power matrix
           into the powerMatrix
    }
}

for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        TransitiveClosure[i][j] +=
            powerMatrix[i][j];
        // taking Union of matrices
        if (TransitiveClosure[i][j] != 0)
        {
            TransitiveClosure[i][j] = 1;
        }
    }
}
} // end of loop that runs (n-1) times
} // end of function
```

Q3) You are given as input two 2-dimensional arrays representing two n-ary relations of degree 10 and 8 respectively (you may assume number of tuples/records in each table by yourself). Write a code fragment to compute J_5 on the input tables.

```
Void ComputeJ5(int R1[5][10], int R2[5][8],
               int J5[5][10+8-5])
{
```

↳ the join will be stored in J5 array passed as parameter

```
    for(int i=0; i<5; i++)
    {
        for(int j=0; j<10+8-5; j++)
        {
            J5[i][j] = -1;   initialize J5 matrix with -1
        }
    }
    int ptr = 0;
    for(int i=0; i<5; i++)
    {
        for(int j=0; j<5; j++)
        {
            if( R1[i][5] == R2[j][0] &&
                R1[i][6] == R2[j][1] &&
                R1[i][7] == R2[j][2] &&
                R1[i][8] == R2[j][3] &&
                R1[i][9] == R2[j][4])
            {
                for(int K=0; K<10; K++)
                {
                    J5[ptr][K] = R1[i][K];
                    // get first 10 fields for J5
                    from 10 fields of R1
                }
```

```
for(int K=10, int l=5 ;
    K< 10+8-5; K++, l++)
{   J5[ptr][K]=R2[j][l];
    // get last 3 fields for
    J5 from last 3 fields
     of  R2
}

ptr++;

j=5;

} // end of IF block.

} // end of inner loop

} // end of outer loop

} // end of function
```

Q4) Let A be the set of students at your school and B the set of books in the school library. Let R1 and R2 be the relations consisting of all ordered pairs (a, b), where student a is required to read book b in a course, and where student a has read book b, respectively. Describe the ordered pairs in each of these relations.

a) $R1 \cup R2$

$R1 \cup R2 = \{(a, b) \mid$ student a is required to read book b or student a has read book b$\}$

b) $R1 \cap R2$

$R1 \cap R2 = \{(a, b) \mid$ student a is required to read book b and student a has read book b$\}$

c) $R1 \oplus R2$

$R1 \oplus R2 = \{(a, b) \mid$ either student a is required to read book b but student a has not read book b or student a has read book b but student a is not required to read book b. $\}$

d) $R1 - R2$

$R1 - R2 = \{(a, b) \mid$ student a is required to read book b but student a has not read book b $\}$

e) $R2 - R1$

$R2 - R1 = \{(a, b) \mid$ student a has read book b but student a is not required to read book b $\}$

(Q5) Let R be the relation on the set of people consisting of pairs (a, b), where a is a parent of b. Let S be the relation on the set of people consisting of pairs (a, b), where a and b are siblings (brothers or sisters).

What are $S \circ R$ and $R \circ S$?

$S \circ R$ is the composition of R and S.

$S \circ R = \{(a, b) \mid$ a is a parent of b and b has a sibling$\}$

$R \circ S$ is the composition of S and R

$R \circ S = \{(a, b) \mid$ a is an aunt or uncle of b$\}$