

南京邮电大学

实验报告

(2024 / 2025 学年 第 一 学期)

课程名称	离散数学			
实验名称	集合上二元关系性质判定			
实验时间	2024	年	10	月 9 日
指导单位	计算机学院计算机科学与技术系			
指导教师	柯昌博			

学生姓名	于明宏	班级学号	B23041011
学院(系)	计算机学院	专 业	信息安全

实 验 报 告

实验名称	集合上二元关系性质判定			指导教师	柯昌博
实验类型	验证	实验学时	4	实验时间	2024.10.9
<p>一、 实验目的和要求</p> <p>为了进一步理解集合上二元关系的性质的判定方法，具体的实验要求如下：</p> <p>输入：可以是具体的二元关系，也可以给定一个具体的集合，随机生成一个二元关系，根据二元关系性质及对应的判定定理，判定是否满足（或不满足）这些性质（自反性，反自反性，对称性，反对称性和传递性）。</p> <p>输出：判定的结果，可以用表格输出来表达此二元关系满足那些性质，不满足那些性质。</p> <p>附加题：求给定二元关系（关系矩阵是高价矩阵）的传递闭包，要求要用 Warshall 算法（算法内容见：教材 124-125 页）求解。</p>					
<p>二、 实验环境(实验设备)</p> <p>硬件：微型计算机</p> <p>软件：Windows 操作系统、Microsoft Visual C++ 2022</p>					

三、实验原理及内容

1、这个程序实现了一个用于生成和分析集合上二元关系性质的工具。在输入关系的元素数量后，随机生成一个关系矩阵，检验其是否满足反身性、非反身性、对称性、非对称性和传递性，并利用 Warshall 算法计算传递闭包。

2、C++源代码：

```
#include <iostream>    // Include the input-output stream library
#include <cstdlib>      // Include the standard library for random functions
#include <ctime>         // Include the library for time functions
using namespace std;   // Use the standard namespace

#define MAX 100        // Define a constant for maximum size of arrays
bool flag_ref, flag_irr, flag_sym, flag_dis, flag_tra; // Flags for properties
int arr[MAX][MAX];     // Matrix to store the relation
int closure[MAX][MAX]; // Matrix to store transitive closure
int n;                 // Number of elements in the relation

// Function to initialize the relation matrix
void initial() {
    cout << "Enter the number of elements (enter 0 to exit):" << endl; // Prompt
user for input
    cin >> n; // Read the number of elements
    if (n == 0) exit(0); // Exit if input is 0
    srand(time(0)); // Seed the random number generator with current time
    // Populate the relation matrix with random 0s and 1s
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < n; ++j)
            arr[i][j] = rand() % 2; // Randomly assign 0 or 1

    cout << endl << "Generated relation matrix:" << endl; // Output the generated
matrix
    // Print the relation matrix
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j)
            cout << arr[i][j] << " "; // Print each element
        cout << endl; // New line after each row
    }
}
```

```

    }

    cout << endl; // Additional new line for formatting
}

// Function to check if the relation is reflexive
void Reflexive() {
    for (int i = 0; i < n; i++)
        if (arr[i][i] == 0) flag_ref = 0; // Check diagonal elements
}

// Function to check if the relation is irreflexive
void Irreflexive() {
    for (int i = 0; i < n; i++)
        if (arr[i][i] == 1) flag_irr = 0; // Check diagonal elements
}

// Function to check if the relation is symmetrical
void Symmetrical() {
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            if (arr[i][j] != arr[j][i]) flag_sym = 0; // Check symmetry condition
}

// Function to check if the relation is dissymmetrical
void Dissymmetrical() {
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            if (arr[i][j] == arr[j][i]) flag_dis = 0; // Check dissymmetry
condition
}

// Function to check if the relation is transitive
void Transitive() {
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)

```

```

        for (int k = 0; k < n; k++)
            if (arr[i][j] && arr[j][k] && !arr[i][k]) flag_tra = 0; // Check
transitivity
    }

// Function to compute the transitive closure using Warshall's algorithm
void Warshall() {
    // Initialize closure matrix with the relation matrix
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            closure[i][j] = arr[i][j];

    // Update the closure matrix based on transitive relations
    for (int k = 0; k < n; k++)
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                closure[i][j] = closure[i][j] || (closure[i][k] && closure[k][j]);
// Transitive step
}

// Function to print the transitive closure matrix
void printClosure() {
    cout << "Transitive closure:" << endl; // Output header for closure
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j)
            cout << closure[i][j] << " "; // Print each element of closure
        cout << endl; // New line after each row
    }
    cout << endl; // Additional new line for formatting
}

// Function to print the properties of the relation
void print() {
    // Print results of reflexivity, irreflexivity, symmetry, dissymmetry, and
transitivity

```

```

        cout << "Reflexivity: " << (flag_ref ? "Satisfied" : "Not satisfied") << endl;
        cout << "Irreflexivity: " << (flag_irr ? "Satisfied" : "Not satisfied") <<
endl;

        cout << "Symmetry: " << (flag_sym ? "Satisfied" : "Not satisfied") << endl;
        cout << "Antisymmetry: " << (flag_dis ? "Satisfied" : "Not satisfied") << endl;
        cout << "Transitivity: " << (flag_tra ? "Satisfied" : "Not satisfied") << endl;
        cout << endl; // Additional new line for formatting
    }

    // Main function to run the program
    int main() {
        while (1) { // Infinite loop to keep the program running
            initial(); // Call the initialization function
            flag_ref = flag_irr = flag_sym = flag_dis = flag_tra = 1; // Reset flags
            Reflexive(); // Check for reflexivity
            Irreflexive(); // Check for irreflexivity
            Symmetrical(); // Check for symmetry
            Dissymmetrical(); // Check for dissymmetry
            Transitive(); // Check for transitivity
            print(); // Print the properties
            Warshall(); // Compute the transitive closure
            printClosure(); // Print the transitive closure
        }

        return 0; // Return 0 to indicate successful completion
    }

```

3、运行结果:

Enter the number of elements (enter 0 to exit):

3

Generated relation matrix:

1 1 0

0 0 0

1 1 1

Reflexivity: Not satisfied

Irreflexivity: Not satisfied

Symmetry: Not satisfied

Antisymmetry: Not satisfied

Transitivity: Satisfied

Transitive closure:

1 1 0

0 0 0

1 1 1

Enter the number of elements (enter 0 to exit):

5

Generated relation matrix:

1 0 0 0 1

1 1 0 0 1

0 1 1 0 1

0 0 0 1 0

0 1 0 0 0

Reflexivity: Not satisfied

Irreflexivity: Not satisfied

Symmetry: Not satisfied

Antisymmetry: Not satisfied

Transitivity: Not satisfied

Transitive closure:

1 1 0 0 1

1 1 0 0 1

1 1 1 0 1

0 0 0 1 0

1 1 0 0 1

Enter the number of elements (enter 0 to exit):

10

Generated relation matrix:

```
0 0 1 1 1 0 1 1 1 0
1 1 0 1 1 0 0 1 0 0
0 1 0 0 0 1 1 0 0 0
0 1 1 0 0 0 1 1 0 0
0 0 1 0 0 0 0 1 0 0
0 1 1 0 0 1 0 1 1 0
0 1 1 0 1 0 1 0 0 1
0 0 1 0 1 1 0 1 1 1
0 1 0 0 1 1 1 1 1 1
1 0 1 1 1 1 0 1 0 0
```

Reflexivity: Not satisfied

Irreflexivity: Not satisfied

Symmetry: Not satisfied

Antisymmetry: Not satisfied

Transitivity: Not satisfied

Transitive closure:

```
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

Enter the number of elements (enter 0 to exit):

0

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

说明：这部分内容主要包括：在编程、调试或测试过程中遇到的问题及解决方法、本次实验的心得体会、进一步改进的设想等。

（一）实验中遇到的主要问题及解决方法

1. 问题：在实现传递闭包时，最初对 Warshall 算法的理解不够深刻，导致代码效率低下。

解决方法：通过查阅资料和参考教材，深入理解了 Warshall 算法的核心思想，优化了嵌套循环的结构，提高了算法的运行效率。

2. 问题：在检查二元关系的性质时，未充分考虑到对称性和反对称性的区别，导致判定结果不准确。

解决方法：结合具体示例，明确对称性和反对称性的定义，调整了逻辑判断条件，确保正确性。

3. 问题：在生成关系矩阵时，随机数生成的分布可能导致某些性质的判定偏差，难以充分覆盖所有情况。

解决方法：设计了多个测试用例，以不同的元素数量和固定的关系模式进行验证，确保全面性。

（二）实验心得

通过本次实验，我对离散数学中集合及其二元关系的性质有了更深刻的理解。编程过程中遇到的问题提升了我的逻辑思维能力，尤其是在调试和优化代码的过程中，更加体会到严谨性的重要性。使用 Warshall 算法计算传递闭包的过程让我对算法设计有了更全面的认识，增强了我在处理复杂问题时的分析能力。

（三）意见与建议（没有可省略）

可以提供更多的时间上机操作，以确保更多程序设计思路得以实现，提升掌握程度和编程能力。

五、支撑毕业要求指标点

支撑毕业要求的指标点为：

- ☐ 1-4 掌握计算机科学与技术领域的专业知识，能将专业知识用于分析和解决计算机领域复杂工程问题。
- ☒ 2-1 能够应用数学、自然科学和工程科学的基本知识，识别和分析计算机领域复杂工程问题的特征。

六、指导教师评语 (含学生能力达成度的评价)

成 绩		批阅人		日 期	
-----	--	-----	--	-----	--

如果不太想写太多字,“指导教师评语”也可以设计为如下的各选择项用打勾形式(仅仅作为一个简单示例, 请各课程负责人根据课程和实验情况以及支撑的指标点来自行设定选择项, 同一门课程的不同实验评分细则项允许存在不同):

	评分项	优秀	良好	中等	合格	不合格
评 分 细 则	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	课题功能实现情况					
	算法设计合理性					
	算法效能评价					
	回答问题准确度					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价 (总成绩)					