



中南大學  
CENTRAL SOUTH UNIVERSITY

# 本科毕业设计(论文)

## GRADUATION DESIGN (THESIS)

通信专业离散数学动态演

题    目    示系统的设计与实现

学生姓名    孙安辉

指导教师    漆华妹

学    院    信息科学与工程学院

专业班级    通信工程 1101 班

本科生院制

2015 年 6 月

# 通信专业离散数学动态演示系统的设计与实现

## 摘要

离散数学(Discrete mathematics)是研究离散量的结构及其相互关系的数学学科,是现代数学的一个重要分支。它在各个学科领域,特别是在计算机科学领域有着广泛的应用。离散数学作为许多计算机相关专业的专业课程,是程序设计语言、数据结构、操作系统、编译原理、数字电路、人工智能、数据库、算法等必不可少的先修课程。对于通信工程专业而言,离散数学也有着重要的地位。

论文通过广泛查阅各种辅助教学软件的设计思想,构建一个离散数学课程的动态演示系统,包括验证公式是否为命题公式、求命题公式的真值表、求两个集合的各种运算、判断关系的性质等模块。所设计的系统能帮助学生理解典型的离散数学知识点,直观的掌握离散数学的部分知识。

**关键词:** 通信工程 离散数学 动态演示系统

# Design and implementation of a dynamic demonstration system of discrete mathematics

## ABSTRACT

Discrete mathematics is a subject which discusses discrete structures and their relations, and it is an important branch of the modern mathematics. It is widely used in various fields, especially in the field of computer science. Discrete mathematics is a specialized courses of many computer related subjects, which is an Advanced Placement class for programming language, data structure, operating system, compiler theory, digital circuit, artificial intelligence, database, and algorithm, etc. As for communication engineering, discrete mathematics also plays an important role.

With widely accessible through a series of auxiliary teaching software design, to build a dynamic discrete mathematics presentation system, which including modules such as checking propositional formula, calculating the truth table, calculating two sets and judging the quality of relations. The designed system can help students understand the typical discrete mathematics knowledge, and intuitively grasp of part of the knowledge of discrete mathematics.

## 目录

第 1 章 绪论 .....	1
1.1 论文研究背景 .....	1
1.2 国内外研究现状 .....	1
1.3 论文的结构 .....	2
第 2 章 系统概要设计 .....	3
2.1 面向对象的编程思想 .....	3
2.1.1 面向对象简介 .....	3
2.1.2 C++ 简介 .....	3
2.1.3 Qt 简介 .....	4
2.1.4 Git 简介 .....	4
2.2 系统设计 .....	5
2.2.1 系统的总体结构 .....	5
2.2.2 系统的演示结构 .....	6
2.3 系统界面设计 .....	7
2.4 系统基本功能 .....	11
2.5 系统主要模块 .....	11
2.5.1 主窗口 .....	11
2.5.2 声明窗口 .....	11
2.5.3 帮助窗口 .....	11
2.5.4 功能模块 .....	12
第 3 章 功能模块分析 .....	13
3.1 验证表达式是否为命题公式 .....	13
3.2 求任意公式的真值表 .....	14
3.3 求任意两个集合的运算 .....	16
3.4 判断关系的性质 .....	17
第 4 章 系统实现 .....	20
4.1 系统概述 .....	20
4.2 主窗口 .....	20

---

4.3 功能模块实现 .....	22
第 5 章 总结与展望 .....	28
5.1 总结 .....	28
5.2 展望 .....	29
致谢 .....	30
参考文献 .....	31

## 第 1 章 绪论

### 1.1 论文研究背景

现今计算机技术已经发展的很成熟了,尤其是基于 windows 平台的可视化软件更是层出不穷。将图形化编程技术运用到教学中,让学生从枯燥的黑板或者幻灯片中解放出来,通过图形形象的去观察,轻松掌握知识点。

离散数学是所有计算机相关专业学生必修的一门基础课。因为离散数学和计算机息息相关。无论是硬件的存储,编码还是操作系统,编译程序等等都离不开它。通信工程中的数字信号处理,信息论与编码,无线通信等等也都会或多或少用到离散数学的知识。所以其在大学学科中的重要地位不言而喻。

但是另一方面,众所周知,这是一门逻辑思维较强、抽象难懂的学科。定义诸多,推理复杂,学生学起来吃力,老师教学的过程中也感觉有难度,对学生掌握的情况也很不满意。传统的教学中,老师播放幻灯片,学生一扫而过,之后什么印象都没有留下。这种授课方式效率不高。

把图形化编程技术和离散数学教学结合起来,实现一个可交互的动态的离散数学演示系统,让学生也可以自己动手操作,老师教授时也增强交互性。学生兴趣高,而且容易直观学习掌握离散数学的关键知识点。

### 1.2 国内外研究现状

网上关于计算机图形化辅助教学工具,还是比较多的。对于计算机学科而言,比较多的有数据结构演示系统<sup>[1]</sup>,算法演示系统等等。但是关于离散数学这门学科的演示系统还是很少见的。国内有部分教师自己实现基于浏览器(javascript,客户端脚本)的一些演示功能,缺点是不完善,而且依赖浏览器。比较著名的有湖南大学杨圣洪的离散数学课程,其实验指导中给出了 javascript 实现的源码和演示界面<sup>[2]</sup>。安徽广播电视大学学报 2013 年第 3 期刊登了邹乐等人的基于 MFC 的离散数学演示系统的设计与实现,包含离散数学中主合取范式、二部图匹配、群的判定、等价关系、等价类等几个子模块,遗憾的是并未给出程序或者源代码<sup>[3]</sup>。

在离散数学的一般教学过程中，都是老师通过板书和播放幻灯片实现。大部分课件都是繁琐的定义和冗长的推理过程。学生听课时很容易昏昏欲睡。

本系统采用面向对象的 C++ 语言开发，直接运行于 windows 平台，携带方便，而且提供了离散数学的动态演示功能。

### 1.3 论文的结构

论文的主要研究工作包括以下几个方面：

- (1) 根据学过的离散数学课程和毕设任务书，对系统的总体轮廓进行设计，得到系统的总体框架。
- (2) 功能理论分析与实现。分析系统要实现的四个功能，每一部分用到的算法，并通过面向对象的方法，用图形化的方式展示出来。
- (3) 设计系统的演示界面。包括主界面，帮助和声明界面。以及具体的每一个功能的演示界面。用户可以随时查看随时使用。
- (4) 编码实现。根据以上步骤的设计和分析，基于 Qt 框架实现源代码

论文的具体组织结构如下：

第 1 章：绪论。简单的介绍了本项目的一些相关情况。

第 2 章：系统概要设计。这部分主要是设计相关。介绍了开发环境和开发工具，然后进行需求分析，设计出系统的模块，以及对系统的界面进行设计。

第 3 章：功能模块分析与实现。这部分主要是实现相关。介绍了四个核心功能模块使用到的算法，以及代码实现的关键。

第 4 章：系统演示。这部分主要是和系统的演示和使用相关。

第 5 章：总结与展望。对本系统的设计与实现进行总结，并提出改进意见。

## 第 2 章 系统概要设计

### 2.1 面向对象的编程思想

#### 2.1.1 面向对象简介

面向对象编程，英文简称 OOP (object oriented programing)。与传统的强调算法的过程性编程不同，它强调的是数据，其理念是设计与问题的本质特性相对应的数据格式。OOP 程序设计的方法首先是设计类，也就是现实事物的一个抽象，他们准确表示了程序要处理的东西。比如设计一个矩形类，该类包含有长和宽两个私有数据成员，并且提供了一些公开的方法（求周长，求面积等等）。在使用该类时，只需要 new 出该类的一个对象，设置好长和宽。直接调用方法即可得到周长和面积。

面向对象，具体的来说涉及抽象、继承、多态等设计方法。如果引申的话，可以理解作为一种设计思想，因为它达到了软件工程的三个主要目标：重用性、灵活性、扩展性，所以这种设计思想已经渗透到软件设计的各个方面，比如窗体编程，数据库编程，游戏开发等等。面向对象的思想又可分为面向对象分析，面向对象设计以及我们经常说的面向对象编程。

采用面向对象的设计方法，优点有很多。代码容易维护，因为采用合理的设计模式后，其程序的设计结构会很清晰，可读性高，而且由于继承的存在，即使改变需求，维护也只发生在局部模块，所以维护起来是方便且低成本的。代码质量高，在设计时，可重用现有的，在以前的项目领域中已被测试过的类来使系统满足业务需求并具有较高的质量。开发效率高，在软件开发时，根据设计的需要对现实世界的事物进行抽象来生成类，使用这样的方法解决问题，接近于日常生活和自然界的思考方式，势必提高软件开发的效率和质量。功能易扩展，由于继承、封装、多态的特性，自然设计出高内聚、低耦合的系统结构，使得系统更灵活、更容易扩展，而且成本较低。

#### 2.1.2 C++ 简介

和 C 语言一样，C++ 也是在贝尔实验室诞生的，Bjarne Stroustrup 于 20 世纪 80 年代在这里开发出了这种语言。C++ 是 C 语言的超集，这意味着任何有效的 C 程序都是有效的 C++ 程序。它们之间有些细微差异，但无足轻重。OOP 部分赋予了 C++ 语言将



问题所涉及的概念联系起来的能力,C 部分则赋予了 C++语言紧密联系硬件的能力。C++正是在 C 的语言基础上添加面向对象的特性,使得该语言更加丰富,并获得了一定程度的成功。

了解过一些编程语言,如 C、C++、java、php 等的同学可能都有体会,C++是最复杂的编程语言之一。事实也的确是这样,因为它要兼 C,所以不能像 java 那样完全面向对象。C++ 有很多知识是让人很容易困惑的,比如引用、多重继承、模板等。但是这并不影响它的杰出特性,代码结构清晰,执行效率高,多种平台支持。C++用于桌面图形化编程是最好不过的了,因为操作系统本身就是使用 C 和 C++编写的。也无需安装其他的运行环境,java 程序就必须安装 jre ,程序才可以跑起来。

### 2.1.3 Qt 简介

Qt 是奇趣科技公司于 1991 年开发的一个跨平台的 C++图形用户界面应用程序框架。它拥有应用程序开发者建立美观的图形用户界面所需要的所有功能。基本上,Qt 同 Windows 平台上的 MFC, ATL 是同类型的东西。Qt 有开源版本,有完整的一套开发环境,编辑器,开发文档,GUI 设计师以及丰富的 API。

Qt 是基于 C++的,采用良好的封装机制使得其模块化程度非常高,可重用性好,对于用户开发来说是非常方便的。而且跨平台,windows 下使用 Qt 写的程序,只需在 Linux 下重新编译,即可实现移植。Qt 开发过程中有两个概念很重要,一个是布局,另外一个信号和槽。布局就是界面设计,水平和垂直方向的组合和对齐等关系。信号和槽则是窗口程序上响应用户操作的关键。

Qt 程序的编译方式有两种静态编译和动态编译。动态编译是缺省的方式,但是这种方法编译出来的可执行文件依赖很多运行库,直接拷贝 exe 文件到其它电脑,程序往往会无法执行。所以这里 Qt 是采用的静态编译的方式安装的,因此项目编译后得到的可执行文件也是静态的(解决了所有库的依赖关系,打包到一个 exe 文件中),可在 windows 平台中任意拷贝使用。

### 2.1.4 Git 简介

首先说明一下开发的过程中为什么要使用版本控制工具。开发的过程中可能会经常遇到这些问题:代码散落各处,整理困难;操作失误,代码无法恢复或者回滚到上一个状态很繁琐;想法改变,难以进行代码重构;多份代码并存,无法确定最新版本。使用

版本控制工具，这些问题就迎刃而解了。版本控制工具又有很多，比如 `cvs`、`svn`、`git` 等。本项目开发采用 `Git`。

`Git` 是一款非常优秀的、开源的、免费的分布式版本控制系统，用于软件开发过程中的源码版本控制，适合应用于各种类型的项目。简单的来说就是管理源代码的各个版本，在不用版本控制工具的时候，我们可能要拷贝整个目录来做个备份，这样一个项目下来可能有好多个目录，而且每一次拷贝备份也没有更多的说明信息。使用 `Git` 就可以避免这样的问题，而且 `Git` 支持远端操作，本地的版本仓库可以同步到远端。这样即使 `word` 崩溃了或是不小心删除了文件，都可以轻松恢复。

`Git` 不仅仅可以用来管理源代码文件，理论上所有的文件都可以。比如论文（这次的论文撰写也使用了 `Git` 做备份，非常方便），程序都是可以的。`Git` 的具体使用可以参考《`Pro Git`》一书。或者廖雪峰的个人网站中关于 `Git` 的教程，该教程浅显易懂，面向初学者，即使没有接触过版本控制的人也可以轻松入门，实用性强，边学边练，一点也不会觉得枯燥。本项目开发过程中，使用 `Git` 做版本控制，同时同步到 [github.com](https://github.com)。

## 2.2 系统设计

### 2.2.1 系统的总体结构

系统总体结构如图 2-1 所示。双击可执行文件，打开主界面后，主窗体有菜单栏（包含关于菜单和查看帮助菜单）、工作区（包含四个操作选项）。

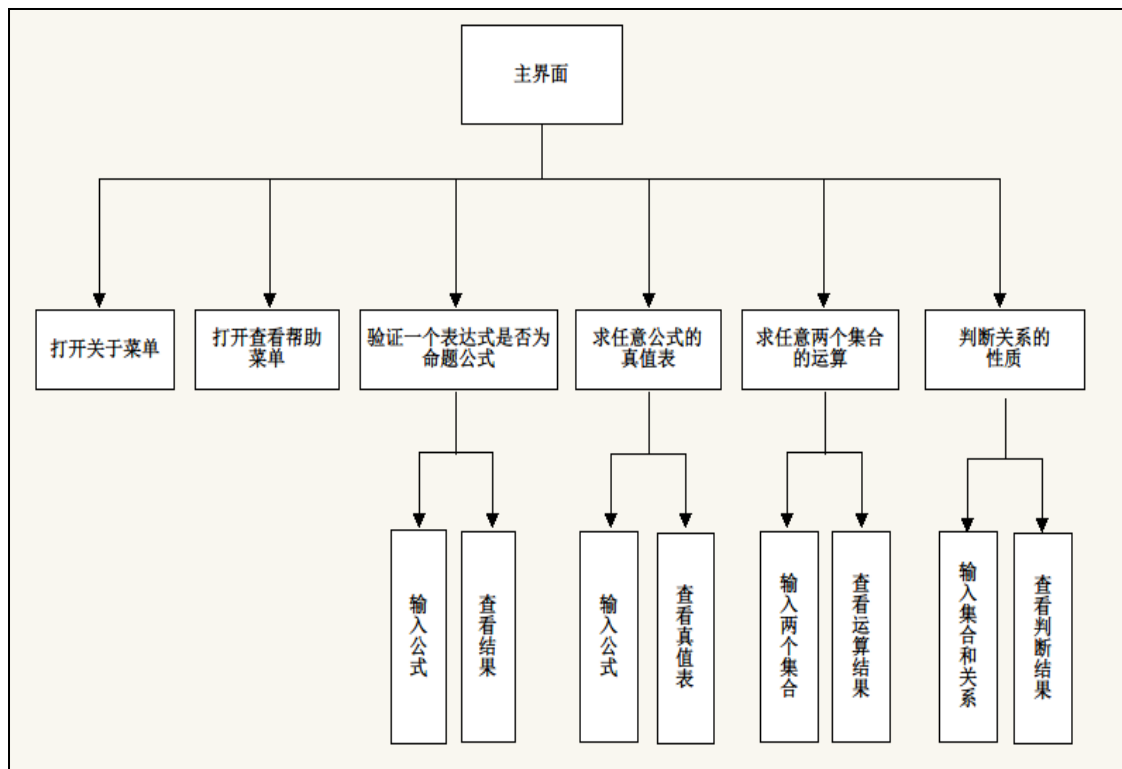


图 2-1 系统总体结构图

### 2.2.2 系统的演示结构

本系统是采用面向对象的设计方法编写的。根据系统需求，整个系统分为两个层次，主界面和演示界面。主界面部分，有菜单栏和四个选项卡，四个选项卡分别对应四个要求的功能。演示界面部分，是系统的核心，用户使用时只需按照界面的提示进行输入。程序经过后台计算自动会把结果显示给用户。

查看帮助菜单提供了一个类似于小型浏览器的功能，可以在其中浏览、跳转、返回，针对每一块都有相应的帮助说明，在使用出现问题的时候即可打开帮助菜单，基本所有的问题都会说明。

演示界面一般大概是由三部分构成。说明部分，简单的说明该离散数学用到的知识点。输入部分，一般是一个文本框，供用户输入使用，在旁边有输入格式提示以及输入确认按钮。结果部分，用户输入完成后按下回车或者确认按钮，系统会将运行的结果以文字或表格的形式反馈给用户。演示界面支持重复使用，每次进行新的输入，无需关闭窗口并重新打开。但是每种演示界面又有所差异，所以可能不全是三部分组成。

整个离散数学演示系统架构简单、层次清晰、操作方便。用户只需点击以及配合键盘进行少量的数据输入，就可达到预期的结果。

### 2.3 系统界面设计

主窗口部分，继承自 `QMainWindow` 类。窗口标题为离散数学动态演示系统。顶部的菜单栏包含两个主选项卡，文件和帮助。文件选项卡下是退出子菜单，支持快捷按键 `Ctrl+X`。帮助选项卡下有两个子菜单，关于和查看帮助。关于窗口主要是版权声明，以及作者介绍和代码地址，继承自 `QDialog` 类。查看帮助菜单窗口是一个类似于浏览器的窗口，使用 `QTextBrowser` 类，主要是针对系统的使用提供简单的帮助。主窗口的主体部分，是四个按钮，每个按钮对应一个核心功能。大概外形如图 2-2 所示。

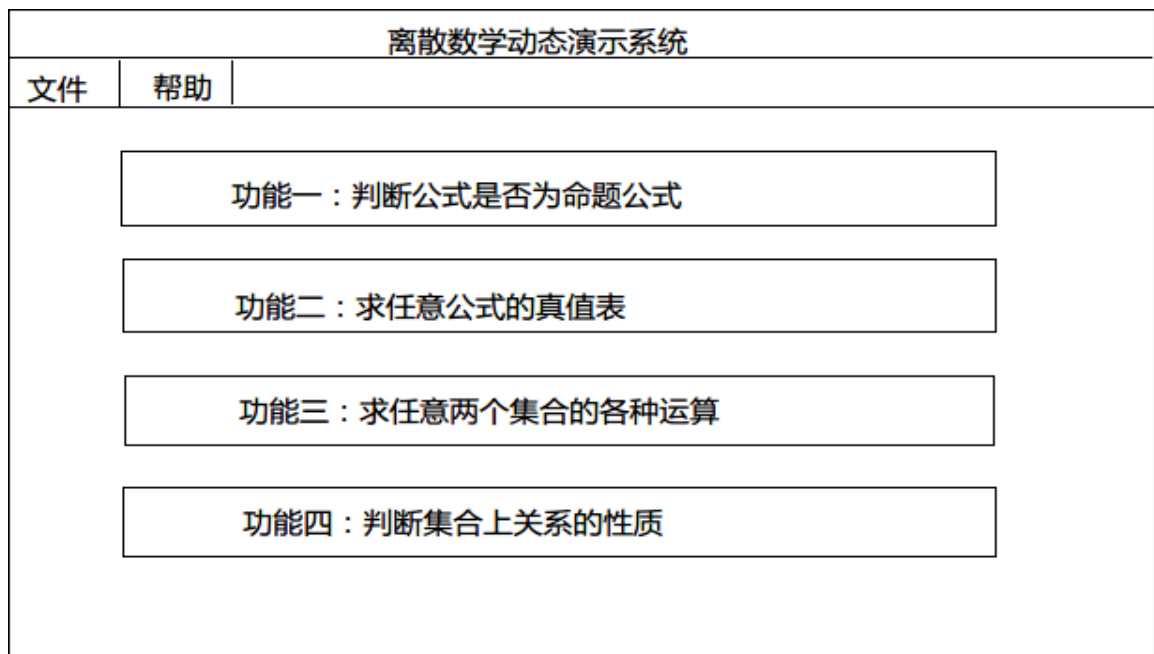


图 2-2 主窗口概念图

验证一个表达式是否为命题公式窗口，继承自 `QDialog` 类。由三个面板构成，首先是说明面板，给出了命题公式的定义。然后是输入面板，有一个输入框，供用户输入公式，旁边是标注，因为命题公式定义中的某些符号，比如合取、析取、蕴含、等值，键盘上是没有这些按键的，所以做了替代。最后是结果面板，会动态的提示用户输入表达式、是命题公式、不是命题公式。大概外形如图 2-3 所示。



图 2-3 判断公式是否为命题公式概念图

求任意公式的真值表窗口，继承自 `QDialog` 类。由两个面板构成，首先是输入面板，有一个输入框，供用户输入命题公式，旁边是标注。这个面板和验证一个表达式是否为命题公式的输入面板外形相似，但是多了输入控制功能。因为只有在用户输入的公式为合法的命题公式的时候，才能求其真值表。所以程序作了控制，如果用户输入了非法公式，会弹出警告框，提示用户重新输入正确的命题公式。然后是结果面板，命题公式的真值表是以表格的形式显示在结果面板中的，由于命题公式长度不一，所以得到的表格大小、外形会有所差异，不过并不影响结果。大概外形如图 2-4 所示。

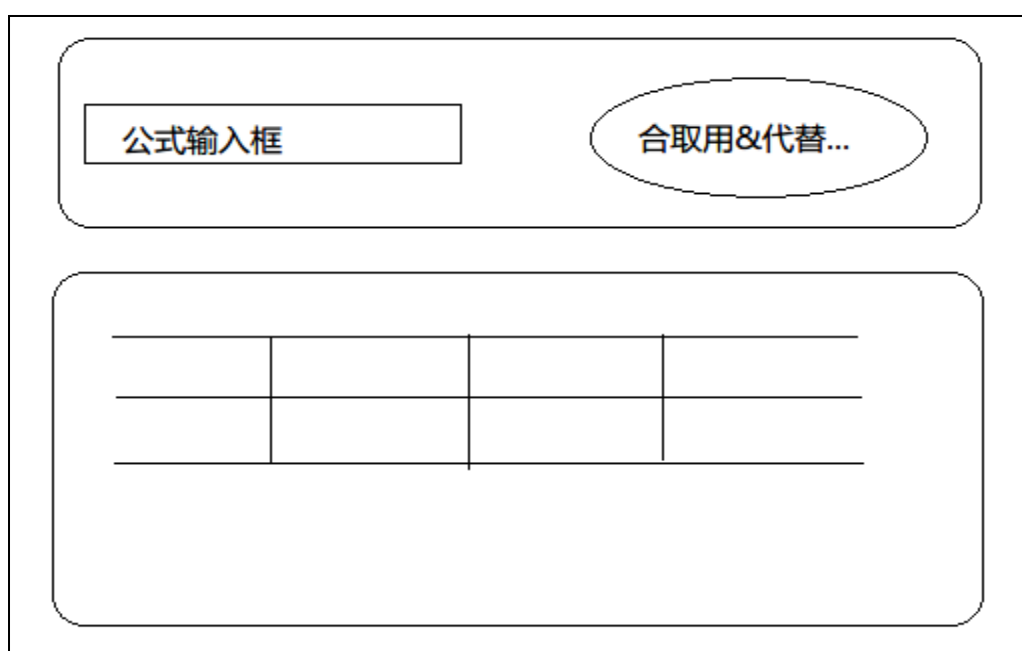


图 2-4 求任意公式的真值表概念图

求任意两个集合的运算窗口，继承自 `QDialog` 类。由四个面板构成，首先是说明面板，给出了集合运算的定义，主要是交集、并集、差集。然后是集合 A 输入面板，用户根据提示的格式要求，输入集合 A。其次是集合 B 输入面板。最后是结果面板，将集合 A 和 B 的交集、并集、差集显示在这里。大概外形如图 2-5 所示。

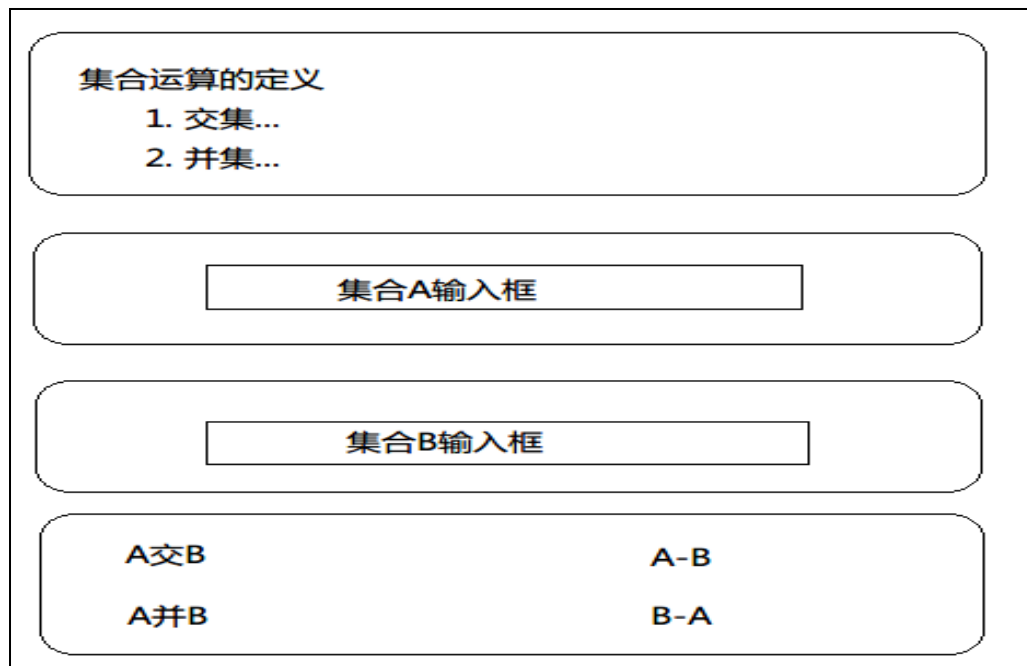


图 2-5 求两个集合的各种运算概念图

判断关系的性质窗口，继承自 `QDialog` 类。由三个面板构成，首先是集合输入面板。然后是关系输入面板。最后是结果面板，程序会将该集合上的关系的性质显示在这里，比如自反、反自反、对称等。大概外形如图 2-6 所示。

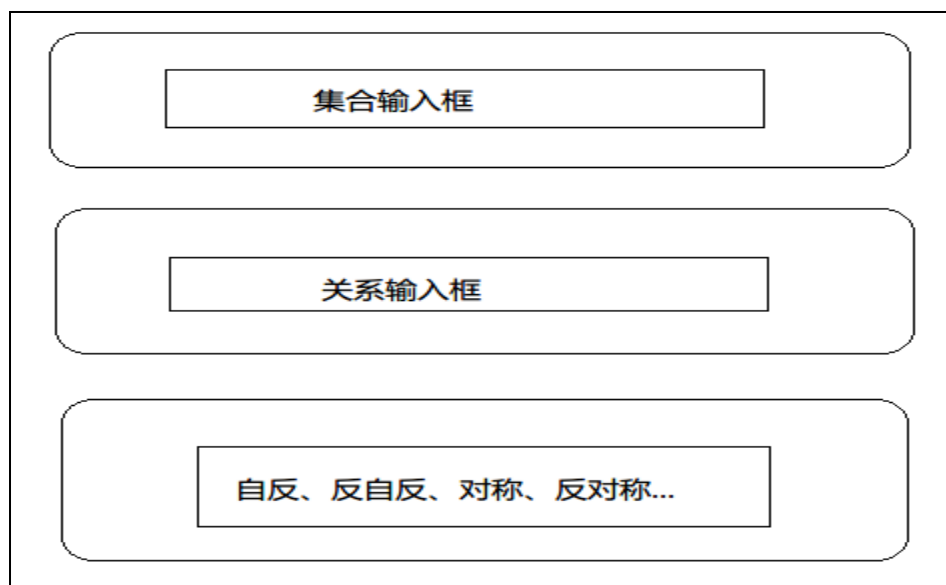


图 2-6 判断关系的性质概念图

界面部分主要是用到 Qt 提供的各种可视化组件类，如 QWidget、QMainWindow、QLabel 等等。可以用 Qt 设计师设计界面，也可以自己写代码控制界面。

## 2.4 系统基本功能

- (1) 提供类似浏览器的帮助功能，用户可以在不同的帮助主题中自由跳转。
- (2) 验证一个表达式是否为命题公式。表达式是任意输入的字符串。
- (3) 求任意公式的真值表。只要公式是合法的命题公式。
- (4) 求任意两个集合的运算。两个集合可以是各种形式的，数字、字母，甚至文字或者空集均可。
- (5) 判断关系的性质。只要输入的集合以及集合上的关系合法。

在上述各个系统功能中，充分展示了图形界面设计的特点，实现人机交互，提高了学生学习的兴趣。只要用户根据系统提示进行操作，无需进行大量的运算画图等，由程序自动给出结果，快速简便。并且演示系统支持扩展，以后如果要添加新的功能，只需在代码中添加新的模块既可。

## 2.5 系统主要模块

### 2.5.1 主窗口

主窗口类继承自 QMainWindow 类，顶部的菜单栏，使用 QMenuBar 类，子菜单使用 QMenu 类，主窗体中提供的四个选项使用的是 QPushButton 类。当用户点击按钮的事件发生的时候，程序响应这些事件所用到的后台机制是信号和槽。简单的理解就是，单击，信号函数会执行，会有相应的槽函数跟着执行，响应该事件。代码编写的时候，一般只需要实现槽函数的功能即可。Qt 窗体程序都是多线程的，这点后台已将帮我们自动维护了，减轻了开发的任务。主窗口主要是集合整个系统的主要功能。

### 2.5.2 声明窗口

声明窗口类继承自 QDialog 类，当用户点击顶部的菜单栏的关于选项时弹出。面板使用 QFrame 类，作者和代码说明信息使用的是 QLabel 类。布局使用的是 QVBoxLayout 类，用于水平布局。声明窗口没有交互功能，用户可以根据其给出的代码地址，自行到网上下载。

### 2.5.3 帮助窗口



帮助窗口类继承自 `QDialog` 类，当用户点击顶部的菜单栏查看帮助选项时弹出。类似浏览器的功能是通过 `QTextBrowser` 类实现，以及上边的返回、主页按钮是通过 `QPushButton` 类实现。帮助窗口的内容可以在几个页面中跳转，分别对应不同的帮助主题，其实每一个页面的实现是一个网页文件，完全使用 `html` 页面编写。这一点将帮助文档独立了出来，如果要修改或者添加帮助文档的内容的时候，只需要去改动 `html` 文件，然后重新编译一下项目就可以了。

#### 2.5.4 功能模块

四个主要的功能模块，因为每一个的逻辑都有所不同，主要是输入、输出结果格式差异很大，所以无法做到统一。窗体界面部分，每一个都继承自 `QDialog` 类，当用户单击相应的功能时弹出。后台处理部分，拿到用户的输入数据，首先进行合法性检查，然后用所对应的算法实现，得到结果后输出在窗口界面。后台代码的实现也是模块化的，针对每一个问题设计一个合理的类，类中提供其算法实现的成员方法。

## 第 3 章 功能模块分析

### 3.1 验证表达式是否为命题公式

要理解离散数学中关于命题公式的定义，先要掌握离散数学中的命题变元和五种基本连接词（也称逻辑运算符），命题变元的表现形式是字母（一般用小写，大写也可以），五种连接词分别是合取（ $\wedge$ ）、析取（ $\vee$ ）、蕴含（ $\rightarrow$ ）、等值（ $\leftrightarrow$ ）、否定（ $\neg$ ）。这五个基本连接词是命题公式的基础元素之一，并且具有不同的优先级，在计算命题公式的值的时候，优先级会起到很大的影响作用。离散数学中对命题公式的定义为：

（1）单个原子公式是命题公式

（2）如果  $A$  和  $B$  是命题公式，则  $(\neg A)$ ， $(A \wedge B)$ ， $(A \vee B)$ ， $(A \rightarrow B)$ ， $(A \leftrightarrow B)$  是命题公式。

（3）只有有限步的应用规则（1）和（2）生成的公式，才是命题公式。

上述这种命题公式的定义为归纳定义，也可以叫做递归定义。通常情况下，我们一般直接判断给定的一个公式是否为命题公式时，就是对公式重复应用以上三条规则，如果符合，则为命题公式，否则就不是命题公式。但是在程序中如果也使用这种递归定义的方法去判断命题公式会非常复杂，所以程序中并没有采取通过递归定义的方式，而是采用的生成公式树的方法去判定。

先举几个简单的例子，看看我们如何判断一个公式是否为命题公式。如给定一个公式  $(P \wedge (P \vee Q))$ 。根据规则（1）， $P$  是命题公式， $Q$  是命题公式。根据规则（2）和前边得到的两个结论， $(P \vee Q)$  是命题公式，再由规则（2），得到  $(P \wedge (P \vee Q))$  是命题公式。再看一个反面的例子， $PQ \wedge R$ ，很显然不能由上述三条形成规则得出，所以不是命题公式。

程序处理的过程如下（输入时合取用  $\&$  代替，析取用  $|$  代替，蕴含用  $-$  代替，等值用  $=$  代替，否定用  $!$  代替，这是因为键盘不能直接输入这些命题连接词）：

（1）获取用户输入的字符串。因为用户输入的字串具有任意性，所以程序对拿到的字符串先进行正则匹配（正则表达式或正则匹配可以理解为对字符串操作的一种逻辑公式，也就是利用预先定义好的一些某些特定字符、及这些特定字符的

组合，形成一个“规则字符串”，然后使用这个“规则字符串”去对目标字符串进行过滤），如果字符串只含有 a-z、A-Z、&、|、!、(、)、-、=、空白，则是合法的字符串，否则直接给出结果：不是命题公式。这个功能的实现可直接使用 Qt 提供的 QRegExp 类，只需要产生该类的一个对象，并用上述规则设定好的字符串初始化对象，然后针对用户输入调用该对象中有关于正则匹配的方法。

(2) 去除字符串中的空格，便于后续处理，因为程序要对字符串中的每一个字符进行判定（是命题变元还是命题连接词），而空格的存在是无意义的。

(3) 构建一棵公式树，树的每一个结点有左右两个孩子。树的父结点为公式中的连接词，树的叶子结点为命题变元 a-z、A-Z。如果可以构建这样的一棵树，那么该公式为命题公式，否则就不是命题公式。构建的过程中涉及连接词的优先级问题，单独调用优先级比较函数判断。

(4) 将结果返回给界面，界面程序会在结果面板给出是否为命题公式的判定。

整个算法的核心在于首先对用户输入的公式进行正则匹配，过滤掉用户输入非法的情况。然后构建这样一棵公式树，叶子结点都为命题变元，非叶子结节点都是命题连接词。公式树构建的过程即是判断公式是否为命题公式的过程。实现的源码文件在 module 目录下的 moduleonefunction.cpp 中。

### 3.2 求任意公式的真值表

命题公式的真假值一般是不确定的。给所有的命题变元赋值，在所有的赋值组合下，得到命题公式取值情况所列成的表称之为真值表。如课本上给出的一个例子：

表 3-1  $((P \vee Q) \wedge P)$  的真值表

P	Q	$((P \vee Q) \wedge P)$
0	0	0
0	1	0
1	0	1
1	1	1

对于含有  $n$  ( $n \geq 1$ ) 个命题变元的命题公式，共有  $2^n$  组赋值。程序的关键就是求得在每一组命题变元的赋值下，命题公式是真还是假（1 或者 0）。然后将得到的每一行数据组合在一起就生成了命题公式的真值表。

程序处理的过程如下（输入时合取用  $\&$  代替，析取用  $|$  代替，蕴含用  $-$  代替，等值用  $=$  代替，否定用  $!$  代替）：

- (1) 获取用户输入的字符串，直接调用上一个模块的方法来判断公式是否为命题公式，进而判断输入的字符串的合法性。如果非法，弹出警告框，用户必须重新输入公式。如果是命题公式，则进行下一步。
- (2) 将命题公式中的命题变元提取出来保存在容器 `vector<char> varNames` 中，因为在构造真值表的时候首行需要列出命题变元。同时调用自己编写的转换方法将命题公式转换成后缀表达式的形式保存起来。所谓后缀表达式就是：不包含括号，运算符（连接词）放在两个运算对象（命题变元）的后面，所有的计算按运算符（连接词）出现的顺序，严格从左向右进行（不再考虑运算符的优先规则）。转化的过程如下：对命题公式按照从左到右的顺序进行扫描；遇到命题变元，加入后缀表达式；遇到连接词，若为左括号，入栈。若为右括号，则依次把栈中的命题连接词从栈中弹出，并加入到后缀表达式中，直到出现左括号为止，同时删除栈中的左括号；若是非括号外的其他连接词，当其优先级高于除左括号以外的栈顶连接词时，直接入栈。否则从栈顶开始，依次弹出比当前处理的连接词优先级高和优先级相等的连接词，直到一个比它优先级低的或者遇到了一个左括号为止。当扫描的命题表达式结束时，栈中的的所有连接词出栈。举个例子，比如  $PVQ\wedge R$ ，转换成后缀表达式就是  $PQR\wedge V$ 。
- (3) 取一组命题变元的赋值组合，对后缀表达式进行计算。计算过程如下：从左到右遍历后缀表达式的每个命题变元和连接词，遇到是命题变元就进栈，遇到是连接词，就将处于栈顶的两个命题变元出栈，进行运算（每一种连接词的运算都预先定义好了），运算结果进栈，一直到遍历结束，最终从栈中获得结果。
- (4) 使用循环，遍历所有的命题变元赋值的组合，得到命题公式的结果，保存在结果容器中。
- (5) 显示界面将变元，命题公式，赋值组合，以及命题公式的值以表格的形式显示在结果面板中。

整个算法的核心在于中缀表达式（原命题公式）到后缀表达式的转换以及对后缀表达式的计算。其中要用到数据结构中的栈（先进后出）结构做辅助，得到的结果由于是

一行行的数据，所以保存在 Qt 提供的容器类模板 `vector` 中，每一个元素对应真值表的一行数据。实现的源码文件在 `module` 目录下的 `moduletwofunction.cpp` 中。

### 3.3 求任意两个集合的运算

离散数学中对集合的定义为：由一个或多个元素所构成的叫做集合。组成集合的每个事物叫做这个集合的元素或成员。通常用大写字母  $A, B, C$  代表集合；用小写字母  $a, b, c$  代表元素。集合的运算定义如下：

设  $A$  和  $B$  是集合，

(1)  $A$  和  $B$  的并集记为  $A \cup B$ ，是集合。

$$A \cup B = \{x | x \in A \text{ 或 } x \in B\}$$

(2)  $A$  和  $B$  的交集记为  $A \cap B$ ，是集合。

$$A \cap B = \{x | x \in A \text{ 且 } x \in B\}$$

(3)  $A$  和  $B$  的差，或  $B$  关于  $A$  的相对补，记为  $A - B$ ，是集合。

$$A - B = \{x | x \in A \text{ 且 } x \notin B\}$$

程序的处理过程如下：

- (1) 获取用户输入的集合（本质上是字符串），将集合中的元素提取出来分别保存在两个集合对象中。
- (2) 对两个集合对象分别使用不同的方法，求其交集、并集、补集。运算的结果保存在几个新的集合对象中。
- (3) 显示界面将集合对象转换成集合的形式，显示在结果面板的相应位置处。

求两个集合的交集运算的算法流程为：建立一个哈希表（`HashTable`），其键（`KEY`）表示集合中元素的值，其值（`VALUE`）表示集合中元素出现的次数。遍历集合  $A$ ，将集合中的每个元素（`KEY`）插入哈希表，每个元素的出现次数（`VALUE`）设置为 1。遍历集合  $B$ ，对于集合中的每个元素：如果哈希表中已经存在该元素，将对应的 `VALUE` 改为 2；如果哈希表中不存在该元素，忽略。遍历哈希表，输出 `VALUE` 为 2 的元素，即得到  $A$  和  $B$  的交集。

求两个集合的并集运算的算法流程为：建立一个哈希表（`HashTable`），其键（`KEY`）表示集合中元素的值，其值（`VALUE`）可以忽略。遍历集合  $A$ ，将集合  $A$  中的每个元素（`KEY`）插入哈希表。遍历集合  $B$ ，对于集合中的每个元素：如果哈希表中已经存在该

元素，忽略。如果哈希表中不存在该元素，将这个元素插入哈希表。遍历哈希表，输出哈希表中的每个 KEY，即得到 A 和 B 的并集。

求两个集合的差集运算的算法流程为：建立一个哈希表（HashTable），其键（KEY）表示集合中元素的值，其值（VALUE）表示集合中元素出现的次数。遍历集合 A，将集合中的每个元素（KEY）插入哈希表，每个元素的出现次数（VALUE）设置为 1。遍历集合 B，对于集合中的每个元素：如果哈希表中已经存在该元素，将对应的 VALUE 改为 2；如果哈希表中不存在该元素，忽略。遍历哈希表，输出 VALUE 为 1 的元素，即得到 A 和 B 的差集。

程序中并未采取上述算法，因为该过程还是比较复杂的。而 Qt 作为一款非常优秀的类型库，不仅支持 C++ 的 STL 模板库，而且提供了大量的自定义“容器类”，专门用于以某种方式存储大量内容。其中就包括集合类 QSet，它的实现就是使用哈希表，即键值对集合。在程序中只需要定义两个 QSet 对象用来保存用户输入的两个集合，然后对这两个对象进行 +、&、-（操作符重载过，即这里的运算以及不是该运算符原有的意义了，而是两个对象之间的运算）操作，分别对应并集、交集、差集，即可得到一个新的集合对象，也就是我们要的结果。实现的源码文件在 module 目录下的 modulethreefunction.cpp 中。

### 3.4 判断关系的性质

离散数学中对二元关系的定义为： $A \times B$  的子集叫做 A 到 B 的一个二元关系。表达有限集合到有限集合的二元关系的时候，关系矩阵和关系图是一有力的工具，因为关系矩阵和关系图不像集合，更加直观，而且特点鲜明。很容易从矩阵和图中看出集合元素之间的关系。集合上关系的性质定义如下：

设 R 是集合 A 上的二元关系，

(1) 如果对 A 中每一 x， $xRx$ ，那么 R 是自反的。即

A 上的关系 R 是自反的  $\forall x (x \in A \rightarrow xRx)$

(2) 如果对 A 中每一 x， $x \not Rx$ ，那么 R 是自反的。即

A 上的关系 R 是自反的  $\forall x (x \in A \rightarrow xRx)$

(3) 如果对每一 x， $y \in A$ ， $xRy$  蕴含着  $yRx$ ，那么 R 是对称的。即

A 上的关系 R 是对称的  $\Leftrightarrow \forall x \forall y (x \in A \wedge y \in A \wedge xRy \rightarrow yRx)$



(4) 如果对每一  $x, y \in A$ ,  $xRy, yRx$  蕴含着  $x=y$ , 那么  $R$  是反对称的。即

$A$  上的关系  $R$  是反对称的  $\Leftrightarrow \forall x \forall y (x \in A \wedge y \in A \wedge xRy \wedge yRx \rightarrow x=y)$

(5) 如果对每一  $x, y, z \in A$ ,  $xRy, yRz$  蕴含着  $xRz$ , 那么  $R$  是传递的。即

$A$  上的关系  $R$  是传递的  $\Leftrightarrow \forall x \forall y \forall z (x \in A \wedge y \in A \wedge z \in A \wedge xRy \wedge yRz \rightarrow xRz)$

关系的性质可能是以上五种中的几种性质的组合（这里限定了输入的集合不能为空集或者空关系，因为空集或空关系的性质基本上是确定的）。程序中无法根据关系性质的定义直接去判断具有何种性质，所以借用了针对不同的关系性质，其二元关系矩阵的特点不同，去判断关系具有哪些性质。

程序的处理过程如下：

- (1) 获取用户输入的集合和集合上的关系。提取集合中的元素保存在数组中。
- (2) 根据集合和集合上的关系的两个数组生成关系矩阵存储在一个二维数组中。
- (3) 依次判断关系矩阵（二维数组）的特点，得到关系的性质。
- (4) 显示界面将步骤 3 得到的性质字符串拼接在一起，然后输出在结果面板中。

关系矩阵是用一个矩阵来表示各个点和每条边之间的关系，也即元素之间的关系。关系矩阵和关系、关系图都是一一对应的。关系矩阵使得关系简化、清晰。

关系具有自反性时，关系矩阵的特点：如果该集合上的关系对应的邻接矩阵的主对角线元素都为 1，则该集合上的关系具有自反性。

关系具有反自反性时，关系矩阵的特点：如果该集合上的关系对应的邻接矩阵的主对角线元素都为 0，则该集合上的关系具有反自反性。

关系具有对称性时，关系矩阵的特点：如果该集合上的关系对应的邻接矩阵具有对称性，即邻接矩阵关于主对角线对称，则该集合上的关系具有对称性。

关系具有反对称性时，关系矩阵的特点：如果该集合上的关系对应的邻接矩阵除主对角线元素之外的任何元素  $R[i][j]=1$  时， $R[j][i]$  必等于 0，则该集合上的关系具有反对称性。

关系具有传递性时，关系矩阵的特点：如果该集合上的关系对应的邻接矩阵当元素  $R[i][j]=1$  且  $R[j][k]=1$  时，必有  $R[i][k]=1$ ，则该集合上的关系具有传递性。

整个算法的核心在于，根据用户输入的集合和集合上的二元关系去构造一个关系矩阵（邻接矩阵），然后根据二元关系矩阵的特点去判断关系所具有的性质。实现的源码文件在 `module` 目录下的 `modulefourfunction.cpp` 中。



## 第 4 章 系统实现

### 4.1 系统概述

离散数学动态演示系统的客户端主要分为两大部分：主界面和模块演示界面。其中模块演示界面是系统的重点和难点。但是这部分用户操作起来是很简单的，只需要按照提示或者帮助菜单中要求的输入格式输入数据，就能得到正确的结果。

### 4.2 主窗口

本系统并未设置登录功能等，用户直接双击 exe 可执行文件，即可进入到主界面中。系统主窗口如图 4-1 所示。顶部的菜单栏处提供了两个主要菜单选项。主窗体部分是对应的四个需求。

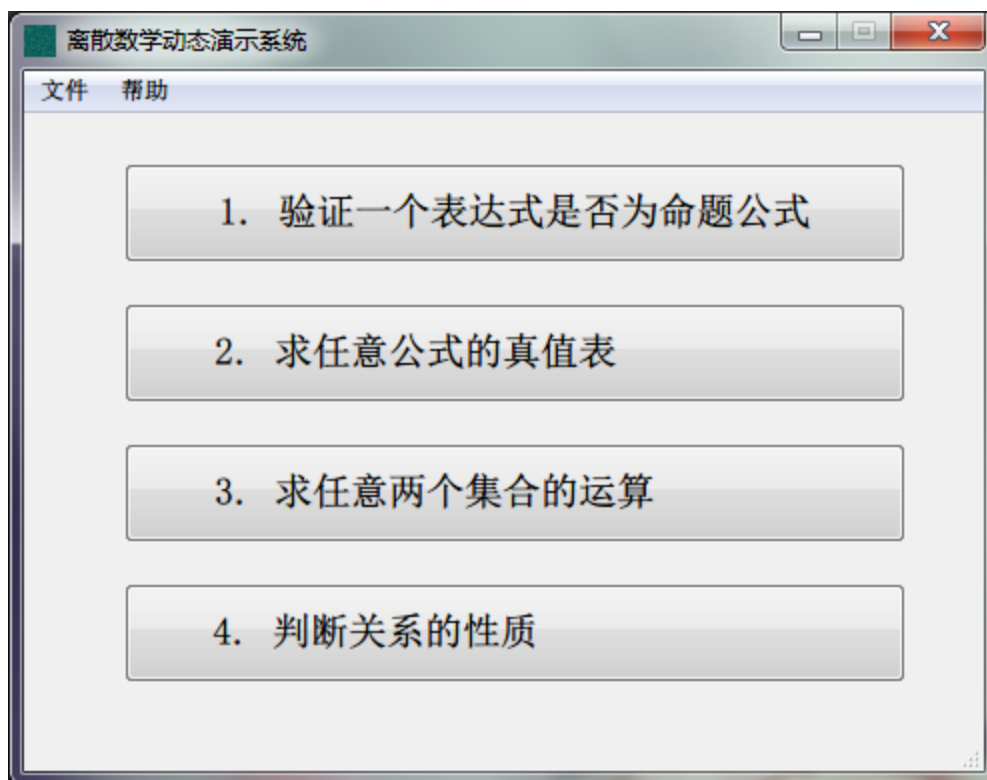


图 4-1 主窗口

菜单栏提供的两个菜单分别是声明窗口和帮助窗口。声明窗口主要是本人的一些信息和项目代码的地址。当用户使用 Git 客户端时，根据给出的地址，即可克隆整个项目源代码到本地。帮助窗口提供了本系统的运行环境说明以及每一个核心功能的使用方法。

其使用方法和浏览器非常相似，可以点击超级链接查看不同的主题，有返回和主页按钮，可以在任意界面中跳转。声明窗口如图 4-2 所示，帮助窗口如图 4-3 所示。

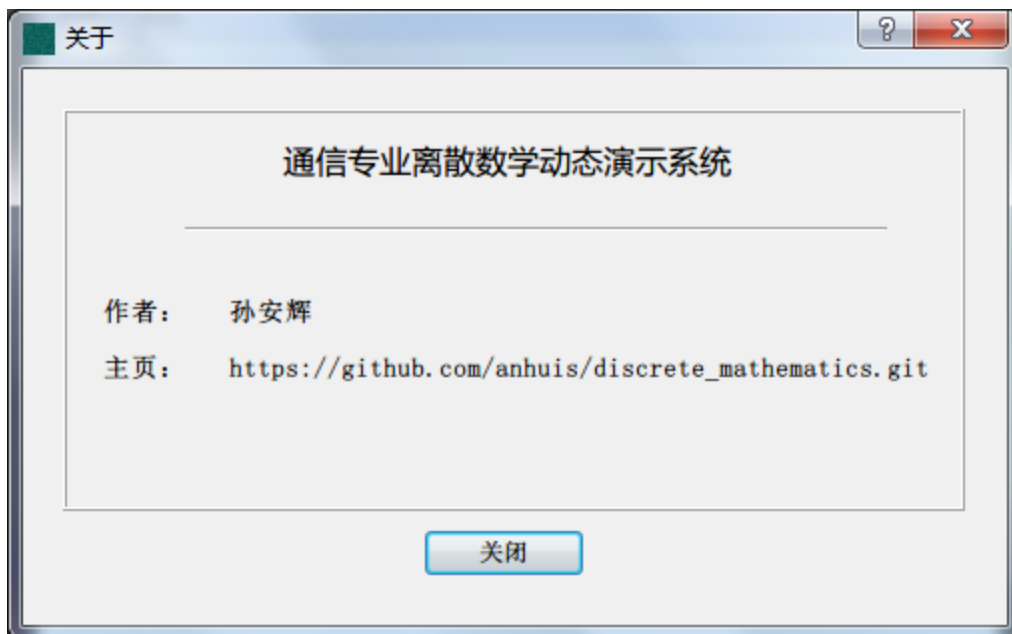


图 4-2 声明窗口

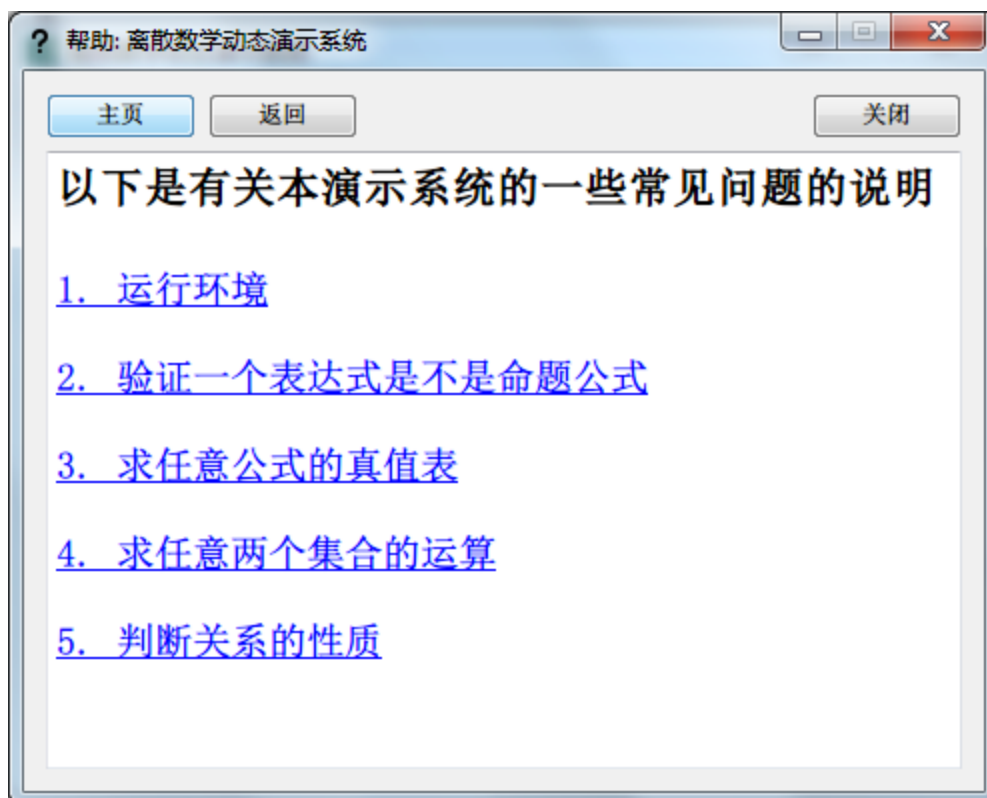


图 4-3 帮助窗口

### 4.3 功能模块实现

该部分是演示系统的核心，分别有四个模块，也即主窗口所显示四个选项。分别是验证一个表达式是否为命题公式、求任意公式的真值表、求任意两个集合的运算、判断关系的性质。用户根据离散数学的知识点选择不同的功能，每一个模块的界面和操作系统均有所不同，以下分别作简单的介绍和分析。

验证一个表达式是否为命题公式。能根据输入的任意字符串判断该公式是不是符合命题公式的定义。程序首先给出命题公式的定义，方便用户理解命题公式。比如我们输入  $P \& Q - R$  ( $P \wedge Q \rightarrow R$ )，这当然是一个命题公式，含有命题变元  $P$ 、 $Q$ 、 $R$  和两个连接词合取和蕴含。其运行如图 4-4 所示。



图 4-4 判断  $P \wedge Q \rightarrow R$  是否为命题公式

再来看一个反面的例子，比如输入  $PQR!$ 。根据命题公式的定义，我们很快可以得出结论，显然这不是一个合法的命题公式，因为否定连接词不能单独存在，命题变元也不可以直接连接在一起。



**验证表达式是否为命题公式**

**命题公式的定义**

- (1) 单个命题变元是个命题公式。
- (2) 若A是命题公式，则 $\neg A$ 是命题公式。
- (3) 若A和B是命题公式，则 $(A \wedge B)$ ， $(A \vee B)$ ， $(A \rightarrow B)$ 和 $(A \leftrightarrow B)$ 都是命题公式。
- (4) 当且仅当有限次地应用(1)，(2)，(3)所得到的含有命题变元、联结词和圆括号的符号串是命题公式。

**输入**

PQR!

**说明：**

合取使用	&	代替	析取使用		代替
蕴含使用	-	代替	等值使用	=	代替
否定使用	!	代替			

**确认**

**结果**

不是命题公式

图 4-4 判断 PQR! 是否为命题公式

求任意公式的真值表。输入任意的命题公式，能将该公式在所有赋值下取值列成一个表，即可得到该公式的真值表。这个功能在离散数学中还是非常的实用，特别是针对命题公式较长也比较复杂的情况下。命题连接词对应有不同的运算规则。否定连接词，0 变 1、1 变 0。合取连接词，只有在全都为 1 的情况下才为 1。析取连接词，在全都为 0 的情况下才为 0。蕴含连接词，只有在前件为 0 后件为 1 的情况下才为 0。等值连接词，只需判断两个是否相等。我们来测试 3.2 节提到的命题公式  $((P \vee Q) \wedge P)$  的真值表，该命题公式只有在 P 为真的情况下才为真，所以只要 P 为 1，命题公式的值就为 1。只需要在输入框输入  $(P|Q)\&P$  即可。其运行情况如图 4-5 所示，和 3.2 节的真值表是完全一致的。我们再测试一组正确的命题公式的数据，比如  $p \wedge q \wedge r$ ，我们自己分析，该命题公式有三个命题变元，应该有 8 种不同的取值组合，并且只有在都赋值为 1 的情况下，命题公式的取值才为 1，其它情况下均为 0。只需要在输入框输入  $p\&q\&r$  即可。其运行情况如图 4-6 所示。当然程序也有错误处理机制，能对用户的错误输入进行提示，比如输入一个非法的公式的时候，程序会弹出警告框，告诉用户重新输入，直到用户输入合法的命题公式为止。

求任意公式的真值表

输入

(P|Q)&P

确认

说明:

合取使用 & 代替

析取使用 | 代替

蕴含使用 - 代替

等值使用 = 代替

否定使用 ! 代替

真值表

	P	Q	(P Q)&P
1	0	0	0
2	0	1	0
3	1	0	1
4	1	1	1

图 4-5 求  $((P \vee Q) \wedge P)$  的真值表

求任意公式的真值表

输入

p&q&r

确认

说明:

合取使用 & 代替

析取使用 | 代替

蕴含使用 - 代替

等值使用 = 代替

否定使用 ! 代替

真值表

	p	q	r	p&q&r
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

图 4-6 求  $p \wedge q \wedge r$  的真值表

求任意两个集合的运算。输入两个集合，可以求出这两个集合的各种运算。程序首先给出集合运算的定义，方便用户随时查看定义。因为集合满足无序性（集合中的元素没有先后之分）、互异性（集合中的元素互不相同）、确定性（集合中的元素必须是确定的）。所以要输入正确的集合元素。比如集合 A 有元素 (1, 2, 3)，集合 B 有元素 (2, 5, 8)。那么 A 和 B 的交集就只有一个元素 (2)。A 和 B 的并集有五个元素 (1, 2, 3, 5, 8)。A 和 B 的差集有两个元素 (1, 3)。B 和 A 的差集有两个元素 (5, 8)。经过以上分析，我们来看程序的运行情况，如图 4-7 所示。

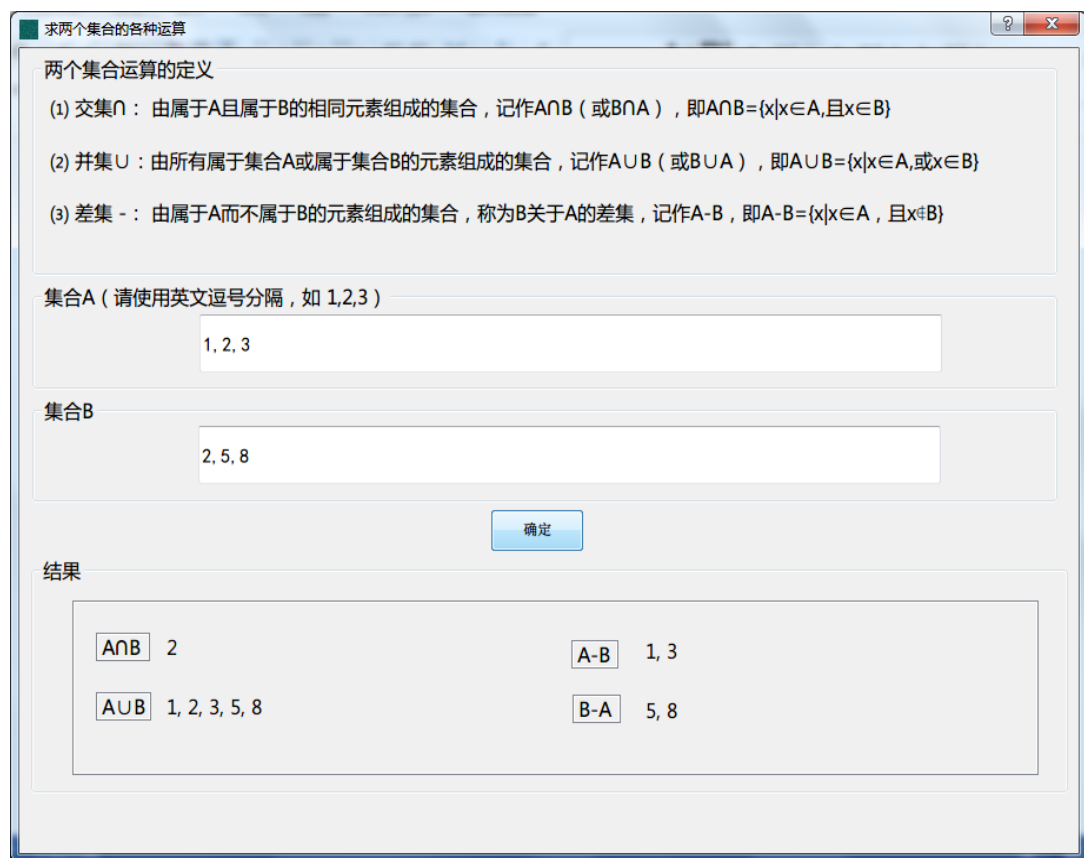


图 4-7 求两个数字集合的各种运算

从程序的运行结果截图来看，完全和我们分析的一致。再举个复杂一点的混合类型集合（集合中的元素有数字，字母）。集合 A 为 (a, A, b, 2, 3)。集合 B 为空集。我们自己分析，A 和 B 的并集应该为 A (a, A, b, 2, 3)，A 和 B 的交集应该为 B (空集)，A 和 B 的差集应该为 A (a, A, b, 2, 3)，B 和 A 的差集应该为 B (空集)。在程序的两个集合输入框，按照提示的要求格式输入集合，需要注意的是空集不需要输入任何东西。程序的运行情况如图 4-8 所示。和我们分析的结果也是一致的。

求两个集合的各种运算

两个集合运算的定义

(1) 交集 $\cap$ ：由属于A且属于B的相同元素组成的集合，记作 $A \cap B$ （或 $B \cap A$ ），即 $A \cap B = \{x | x \in A, \text{且} x \in B\}$

(2) 并集 $\cup$ ：由所有属于集合A或属于集合B的元素组成的集合，记作 $A \cup B$ （或 $B \cup A$ ），即 $A \cup B = \{x | x \in A, \text{或} x \in B\}$

(3) 差集 $-$ ：由属于A而不属于B的元素组成的集合，称为B关于A的差集，记作 $A - B$ ，即 $A - B = \{x | x \in A, \text{且} x \notin B\}$

集合A（请使用英文逗号分隔，如 1,2,3）

a, A, b, 2, 3

集合B

确定

结果

$A \cap B$

空集

$A - B$

2, 3, A, a, b

$A \cup B$

2, 3, A, a, b

$B - A$

空集

图 4-7 求两个特殊集合的各种运算

判断关系的性质。输入一个集合和该集合上的一个二元关系，可判断出该关系的性质（是否为自反、反自反、对称、反对称、传递关系）。例如，集合  $A = \{1, 2, 3\}$ ，该集合上的关系  $R = \{\langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle, \langle 1, 2 \rangle\}$ 。我们列出它的关系矩阵，很容易发现对角线的元素全为 1，所以该集合上的关系是自反的。也满足反对称的特点（除主对角线元素之外的任何元素  $R[i][j]=1$  时， $R[j][i]$  必等于 0）。同时也满足传递性的特点，这点可根据关系图，很容易判断。该例子的关系矩阵如图 4-8，关系图如 4-9。程序运行情况如图 4-10。

1	1	0
0	1	0
0	0	1

图 4-8 关系矩阵

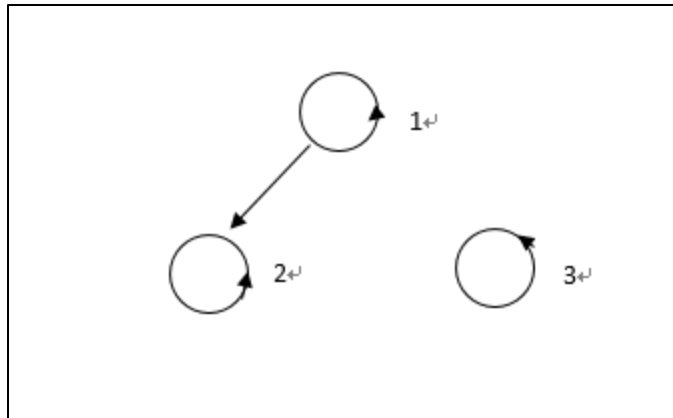


图 4-9 关系图



判断关系的性质

集合 (请使用英文逗号分隔, 如 1,2,3 )

1, 2, 3

关系 (请按照以下格式输入: <1,2>,<2,3>,<1,3> )

<1, 1>, <2, 2>, <3, 3>, <1, 2>

确定

结果

该关系具有: 自反性 反对称性 传递性

图 4-10 判断关系的性质

从程序的运行结果看, 该集合上的关系具有自反性、反对称性和传递性。这和我们通过该集合上的关系的关系矩阵以及关系图得到的分析结果是一致的。



## 第 5 章 总结与展望

### 5.1 总结

离散数学作为一门计算机相关专业的基础课，重要性不言而喻。但是其具有太多的定义和演算过程。在教学和学习的过程中，不免会感到有些吃力。本次设计的离散数学动态演示系统，一定程度上改进了这些问题。本系统同时适合学生和老师使用，比如要展示一个命题公式的真值表，直接使用系统提供的功能，就能快速得到命题公式的真值表，节约了在黑板演算和画表格的时间。

本系统采用面向对象的编程语言 C++ 开发，基于跨平台的 Qt 框架，使用开源的 Git 做版本控制工具，编译环境为 32 位 windows 7 操作系统。演示系统界面主要分为主界面和演示界面两大部分。主界面组织整个软件的功能。演示界面一般包括三部分，离散数学知识点相关的说明部分，用户数据输入部分，结果展示部分。系统支持一定程度的人机交互，用户每次不同的输入，结果区都会有不同的结果显示。在输入部分还会提示用户的输入是否有误，帮助用户正确输入数据。本系统采用静态编译的方式生成，携带方便，运行时无需安装格外的运行环境，直接可以运行于各类 windows 平台。

整个项目开发的过程中，经历了项目需求分析、系统概要设计、系统界面设计、核心算法实现、编码实现、调试发布等过程。通过这次的软件设计与实现，让我学到了很多。首先，工欲善其事，必先利其器，前期的准备工作无疑是很重要的，包括编程语言的基础（主要是 C++）、Qt 框架的使用、Git 版本控制的使用以及重要的数据结构。然后就是软件的设计阶段，从宏观的设计整个软件的运行界面，需要实现的后台功能。接下来是代码的编写，这个过程可以说是有点痛苦和枯燥的，需要不断的调试，不断的查阅大量的英文开发文档，但是当完成一个功能的时候，也很有成就感。最后就是软件的测试阶段了，修改 bug，完善注释。

通过这次毕业设计项目开发，拓宽了我的知识面，锻炼了自己的分析问题和编码的能力，综合素质得到了较大提高。总之，这两个多月的开发过程，自己收获了很多。也对软件开发有很深的体会，软件领域中的知识在于积累，问题决定解决方案。不明白的问题，要多上搜索引擎问和查阅 API 文档。还有就是如果有可能，一定要写好注释和开发文档。

## 5.2 展望

由于本人的水平有限以及时间不是很充足的因素，本演示系统虽然完成了需求中要求的基础的演示功能，但是演示系统还有很多不足的地方，需要继续改善。主要体现在以下方面：

- (1) 系统界面。设计相对简单，运行时看起来也不够美观。作为一款教学软件，严谨性略显不足。菜单栏可以增加更多的菜单，主界面也可以设置背景图片。底部的状态栏如果能实时给出简单的运行提示会更好。每一个功能的演示界面，元素也比较单一，主要是文字和输入框，可以加以美化。
- (2) 算法实现方面。四个核心模块的算法大部分是从最简单的入手，首要考虑的是完成该功能，从而并未考虑很多运行效率方面的问题。如果能采用多样化的后台算法实现，然后进行横向比较再去选择性能最优的会更好。
- (3) 注释方面。源码的注释还是相对较少，只有在算法部分加了简单的注释，这导致后来维护代码的时候自己也有部分一下不是很懂。所以要完善注释。
- (4) 功能方面。演示系统并未提供支持单步演示的功能。主要是考虑到有些离散数学的运算，单步演示支持也没有意义，比如判断一个公式是否为命题公式，程序中是通过构建公式树去判断的，将这种判断过程展现给用户是毫无意义的，因为使用者可能根本不知道什么是数据结构中的树结构。但是另一方面，可以在功能上完善一下，比如判断关系的性质，可以在界面上给出关系的关系矩阵和关系图。再比如求真值表的时候，可以将命题公式的真值表输出到文件中保存起来，方便下次查阅。
- (5) 权限管理方面。作为一款教学软件，如果加以权限限制，不同账号提供不同的功能，会更完善。
- (6) 扩展方面。最好能支持用户自定义模块，当用户学完相关知识后，可以自己设计演示功能并添加到系统中。

本项目已经放到 `github` 平台（一个 `Git` 库的托管网站，简单理解就是把项目放到该网站保存起来，包含有项目开发的每一个版本），感兴趣的开发者可以参与到项目中来，一起完善。`github` 项目地址为 [https://github.com/anhuis/discrete\\_mathematics.git](https://github.com/anhuis/discrete_mathematics.git)。也欢迎大家下载源代码编译运行，并给出指正。

## 致谢

毕业论文能够按时完成，首先要感谢我的导师漆华妹老师。从最开始的毕设进度安排到之后的每周进度检查 and 中期检查，漆老师总是不厌其烦的帮我们看邮件、审阅我们提交的进度报告，解决我们遇到的各种疑问。早在上学期期末的时候，漆老师就安排我们做英文文档的翻译和着手准备调研报告的编写，所以我们组的进度比其他组要快多了，也才能在这学期有更多的时间编写软件和撰写论文。在中期答辩的时候，由于我自身的问题，出了点小麻烦。还好在漆老师耐心的帮助下，帮我解决了问题，顺利通过了论文的中期检查。所以，在这里向老师表示衷心的感谢。

同时还要感谢我的同学和室友，是你们陪我渡过了大学四年最美好的时光，让我在外地求学的路上不孤单。大学最后这一学期，我和室友们在一起写论文，遇到论文编写的格式问题、论文提交的时间问题、查重问题等等，都会一起讨论，有时还帮我搜集资料。你们给了我很大的鼓励，让我拥有信心去解决这段时间遇到的各种问题。

还要特别感谢我的姐姐，在完成论文的过程中，她用自身本科毕业时的经历给了我很多指导，帮我收集素材和解决论文的排版问题。再者感谢我的父母，在我十六年的漫长异地求学路上，是你们在背后的默默付出和对我的支持，让我得以完成学业。

回想从大一到大四这四年年在中南的生活，有欢笑也有泪水，真可谓酸甜苦辣咸五味陈杂。在学校的这些美好时光也很快会成为记忆。所以很感谢母校，四年来让我学到了很多知识，从思想上、生活上、学习上、为人处世上都得到了很大的成长。这四年，给我的青年时代画上了完美的句号，但这不是结束，而是一个新的开始，它是我一辈子都受用不尽的财富。

祝所有我感谢的人们，一生幸福平安。

## 参考文献

- [1] 严蔚敏. 数据结构[M]. 北京: 清华大学出版社, 2001, 4~17.
- [2] 湖南大学杨圣洪离散数学课程实验指导  
<http://kczx.hnu.cn/G2S/Template/View.aspx?courseId=556&topMenuId=126997&action=view&type=&name=&menuType=1&curfolId=122835>
- [3] 邹乐, 华姗姗, 吕刚. 基于 MFC 的《离散数学》实验演示系统的设计与实现. 安徽广播电视大学学报, 2013.
- [4] 吴红梅, 罗国明. C 程序设计教学中动态演示系统的实现. 信息通信, 2013, (8): 97-97.
- [5] 徐静. 体验项目驱动的 JAVA 程序设计课程建设思路. 计算机时代, 2014, (5): 64-65.
- [6] Jasmin Blanchette, Mark Summerfield. C++ GUI Qt 4 编程 (第二版). 北京: 电子工业出版社, 2013.
- [7] Alan Ezust, Paul Ezust. C++ Qt 设计模式 (第二版). 北京: 电子工业出版社, 2012.
- [8] Scott Chacon. Pro Git. 美国: Apress, 2009.
- [9] 廖雪峰 Git 教程.  
<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>
- [10] Kenneth A. Reck. C 和指针. 北京: 人民邮电出版社, 2012.
- [11] LinDen. C 专家编程. 北京: 人民邮电出版社, 2008.
- [12] Stephen Prata. C++ Primer Plus (第五版) 中文版. 北京: 人民邮电出版社, 2012.
- [13] 方世昌. 离散数学 (第三版). 西安: 西安电子科技大学出版社, 2010.
- [14] 赵曦. 《编译原理》CAI 动态演示系统的研究与研发. 教育教学论坛, 2013, (43): 157-158.
- [15] 李强, 明艳, 吴坤君. 基于 Matlab 的通信原理演示系统的设计与应用. 实验技术与管理, 2008, 25 (8): 77-79.
- [16] 谭励, 姜同强. 离散数学课程实验与互动平台建设. 时代教育, 2014, (17): 138-139.
- [17] 路芳. 在离散数学中引入研究型题目的教学实践. 阴山学刊: 自然科学版, 2014, 28 (1): 86-88.
- [18] 孙海民. 设计开发“数据结构”教学软件的现状分析[J]. 计算机教育, 2005, 10: 22~27.