



同濟大學  
TONGJI UNIVERSITY

# 《离散数学》课程实验报告

题目 命题逻辑推理

姓 名 林继申

学 号 2250758

学 院 软件学院

专 业 软件工程

教 师 李 冰

二〇二三年十二月二十二日

# 目录

1 实验目的.....	1
2 实验内容.....	1
3 实验环境.....	1
4 实验原理.....	2
4.1 合取.....	2
4.2 析取.....	2
4.3 条件.....	2
4.4 双向条件.....	3
4.5 命题逻辑推理.....	3
5 实验过程.....	3
5.1 实验思路.....	3
5.2 实验设计.....	4
5.2.1 数据结构设计.....	4
5.2.2 结构体与类设计.....	4
5.2.3 程序主体架构设计.....	4
5.3 核心算法实现.....	5
6 实验数据分析.....	6
7 实验心得.....	6
8 程序源文件.....	7

## 1 实验目的

本实验的主要目的是加深对命题逻辑推理方法的理解。命题逻辑是数学逻辑的一个基础分支，它涉及使用形式化方法来表达逻辑语句和推断。通过本实验，学生将能够：

- (1) 掌握基本的命题逻辑概念，包括命题、联结词、逻辑等价、蕴含等；
- (2) 学习如何将日常语言中的语句转化为命题逻辑表达式；
- (3) 理解和应用基本的逻辑运算，包括合取、析取、否定、条件、双条件等；
- (4) 发展逻辑思维和问题解决技巧，特别是在处理逻辑推理问题时；
- (5) 通过编写程序，实践命题逻辑在计算机科学中的应用。

实验通过对具体逻辑推理问题的分析和编程解决，使学生能够将理论知识与实践应用相结合，提升逻辑思维和编程能力。

## 2 实验内容

实验内容为用命题逻辑推理的方法解决逻辑推理问题。问题描述：根据下面的命题，用命题逻辑推理方法确定作案者，并给出推理过程。

命题 1：营业员 A 或 B 偷了手表

命题 2：若 A 作案，则作案不在营业时间

命题 3：若 B 提供的证据正确，则货柜未上锁

命题 4：若 B 提供的证据不正确，则作案发生在营业时间

命题 5：货柜上了锁

将上述命题用命题变元表示：

命题 A：营业员 A 偷了手表

命题 B：营业员 B 偷了手表

命题 C：作案不在营业时间

命题 D：B 提供的证据正确

命题 E：货柜未上锁

## 3 实验环境

程序开发语言：C++

集成开发环境：Microsoft Visual Studio 2022 (Release 模式)

编译运行环境：本项目适用于 x86 架构和 x64 架构

## 4 实验原理

### 4.1 合取

合取是一种二元逻辑运算。将两个命题  $P$ 、 $Q$  联结起来，构成一个新的命题  $P \wedge Q$ ，读作  $P$ 、 $Q$  的合取，也可读作  $P$  与  $Q$ 。当且仅当两个命题都为真时，它们的合取才为真。如果其中任何一个命题为假，它们的合取就为假。

表 4.1.1 合取运算真值表

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

### 4.2 析取

析取是一种二元逻辑运算。将两个命题  $P$ 、 $Q$  联结起来，构成一个新的命题  $P \vee Q$ ，读作  $P$ 、 $Q$  的析取，也可读作  $P$  或  $Q$ 。当至少一个命题为真时，它们的析取就为真。只有当两个命题都为假时，它们的析取才为假。

表 4.2.1 析取运算真值表

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0

### 4.3 条件

条件是一种二元逻辑运算，也称为蕴含。将两个命题  $P$ 、 $Q$  联结起来，构成一个新的命题  $P \rightarrow Q$ ，读作  $P$  条件  $Q$ ，也可读作如果  $P$ ，那么  $Q$ 。如果第一个命题（前件）为真，则第二个命题（后件）也必须为真，整个条件才为真。唯一使条件为假的情况是前件为真而后件为假。

表 4.3.1 条件运算真值表

P	Q	$P \rightarrow Q$
1	1	1
1	0	0
0	1	1
0	0	1

## 4.4 双向条件

双向条件是一种二元逻辑运算，也称为等价。将两个命题  $P$ 、 $Q$  联结起来，构成一个新的命题  $P \leftrightarrow Q$ ，读作  $P$  双条件于  $Q$ ，也可读作  $P$  等价  $Q$ 。当两个命题具有相同的真值时（要么都为真，要么都为假），它们的双向条件为真。

表 4.4.1 双向条件运算真值表

P	Q	$P \leftrightarrow Q$
1	1	1
1	0	0
0	1	0
0	0	1

## 4.5 命题逻辑推理

命题逻辑推理是一种基于形式化命题和逻辑运算的推理系统，主要用于分析和推导陈述的真值。在这个系统中，每个命题被定义为一个可以判定为真或假的陈述，而通过逻辑运算符如“和”、“或”和“非”，这些命题可以组合成更复杂的表达式。命题逻辑的核心在于通过系统化的方法推导出从一组假设到结论的逻辑关系，使其在数学证明、计算机科学等领域得到了广泛应用。

# 5 实验过程

## 5.1 实验思路

在本实验中，目标是运用命题逻辑推理来解决一个逻辑难题，即确定谁是作案者。实验的核心思路是首先将问题中的每个陈述转换成逻辑命题，并为每个命题分配一个变量。接着，利用 C++ 编程，结合逻辑运算符如“与”、“或”和“非”，来构建一个包含所有这些命题的复合逻辑表达式。程序通过遍历所有可能的命题变量组合，找到使得整个复合命题为真的特定组合，从而明确指出作案者的身份。

## 5.2 实验设计

### 5.2.1 数据结构设计

由于这个程序主要涉及基本的逻辑运算，因此不需要复杂的数据结构。主要使用基本的数据类型（如布尔值）来存储命题变元的真值。

### 5.2.2 结构体与类设计

在这个实验中，没有使用结构体或类，因为程序的规模和复杂性不需要这样的封装。

### 5.2.3 程序主体架构设计

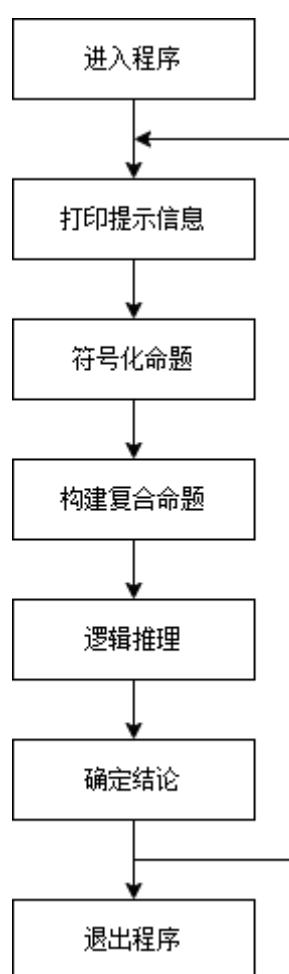


图 5.2.3.1 程序主体架构设计流程图

程序主体架构设计如下：

(1)符号化命题：首先，将问题描述中的命题转化为命题逻辑表达式。这包括对每个关键元素分配一个逻辑变量（如 A 代表“营业员 A 偷了手表”）。这一步骤是建立清晰逻辑框架的基础，确保每个命题都可以在逻辑运算中得到正确的表示；

(2)构建复合命题：利用逻辑运算符（如与、或、非）将这些基本命题结合成一个复合命题。这个复合命题将包含全部的逻辑关系，使我们能够从中推导出最终结论；

(3)编写程序以实现逻辑推理：在程序中，我们使用了一个循环来遍历所有可能的真值组合（共 32 种，因为有 5 个独立的命题变元，每个有两种状态）。这一过程涉及到逻辑运算，如“与”和“或”，以及条件语句，来判断哪些组合满足我们的复合命题；

(4)确定结论：程序通过检查所有可能的命题变元组合，找到满足所有条件的特定组合。这一步是逻辑推理的关键，它将直接指向哪位营业员是作案者。在这个特定的问题中，程序将输出满足条件的命题变元的真值状态，从而揭示作案者的身份；

(5)用户交互：程序提供了清晰的用户界面和指示，引导用户理解问题描述和命题变元的表示方法。同时，程序在完成逻辑推理后输出结果，并等待用户按键以结束程序。

### 5.3 核心算法实现

在程序中，我们使用了一个循环来遍历所有可能的真值组合（共 32 种，因为有 5 个独立的命题变元，每个有两种状态）。这一过程涉及到逻辑运算，如“与”和“或”，以及条件语句，来判断哪些组合满足我们的复合命题。核心算法的实现过程如下：

(1)遍历所有可能的命题组合：程序使用 for 循环来遍历所有可能的命题真值组合。因为有五个独立的命题 (A,B,C,D,E)，每个命题有两种可能的状态（真或假），总共有  $2^5=32$  种组合。循环变量 i 从 0 到 31（包括 0 和 31），每个值代表一种独特的命题组合；

(2)命题变量的赋值：使用位运算来确定每个命题的真值状态。每个命题 (A,B,C,D,E) 都与一个特定的位相关联。例如， $i\&0b1$ （二进制的 1）用来确定 A 的状态， $i\&0b10$ （二进制的 10）用来确定 B 的状态，依此类推。这些位运算根据 i 的当前值，通过检查相应的位是 0 还是 1，分别为每个命题赋予假或真的值；

(3)逻辑表达式的计算：接下来，算法评估一个复合逻辑表达式： $if((A||B)\&\&(!A||C)\&\&(!D||E)\&\&(D||!C)\&\&!E)$ 。这个表达式结合了与、或和非逻辑运算符，代表了问题中描述的所有条件。每个条件反映了问题陈述中的一个命题，例如，“若 A 作案，则作案不在营业时间”转化为逻辑表达式  $!A||C$ 。这个复合逻辑表达式只有在所有的条件同时满足时才为真；

(4)确定并输出结果：当找到满足所有条件的命题组合时，程序进入 if 语句块，并打印出哪位营业员 (A 或 B) 偷了手表。输出包括每个变量的真值状态，

以及最终的结论：“所以营业员 A/B 偷了手表”。

## 6 实验数据分析

```
+-----+
|           命题逻辑推理           |
| Propositional Logic Reasoning    |
+-----+

>>> 问题描述
[命题 1] 营业员 A 或 B 偷了手表
[命题 2] 若 A 作案，则作案不在营业时间
[命题 3] 若 B 提供的证据正确，则货柜未上锁
[命题 4] 若 B 提供的证据不正确，则作案发生在营业时间
[命题 5] 货柜上了锁

>>> 命题变元表示
[命题 A] 营业员 A 偷了手表
[命题 B] 营业员 B 偷了手表
[命题 C] 作案不在营业时间
[命题 D] B 提供的证据正确
[命题 E] 货柜未上锁

>>> 命题逻辑推理结果：A = False, B = True

>>> 所以营业员 B 偷了手表

Press Enter to Quit
```

图 6.1 实验数据

实验数据分析：程序通过检查所有可能的命题变元组合，找到满足所有条件的特定组合。命题逻辑推理结果：为  $A = \text{False}$ ,  $B = \text{True}$ 。所以营业员 B 偷了手表。

## 7 实验心得

通过实验，我加深了对命题逻辑基础概念的理解，包括命题、联结词、逻辑等价、蕴含等。这不仅增强了我的逻辑思维能力，还提升了我的问题解决技巧，特别是在处理逻辑推理问题时。

实验的过程中，我学会了如何将日常语言中的语句转化为命题逻辑表达式，并运用合取、析取、否定、条件、双向条件等基本逻辑运算。这些技能对于分析复杂问题和构建有效算法至关重要，尤其在计算机科学领域。

编写 C++ 程序来解决逻辑推理问题的过程是本次实验的一个亮点。我通过程序实现了对逻辑命题的推理，成功地运用了逻辑运算符，并处理了复杂的条件语



句，这个过程检验了我的编程技能。

通过本次实验，我体验到了理论与实践相结合的重要性。我不仅学到了逻辑推理的理论知识，还通过实际编程实践了这些理论。这种结合理论与实践的学习方式极大地增强了我的学习效果，让我对离散数学有了更深的认识和兴趣。

## 8 程序源文件

```
/*
*****
* Project Name: Assignment_2
* File Name: assignment_2.cpp
* File Function: 命题逻辑推理
* Author: Jishen Lin (林继申)
* Update Date: 2023/12/13
*****
*/

#include <iostream>
#include <conio.h>
#include <cmath>

/*
* Function Name: main
* Function: Main function
* Return Value: 0
*/
int main()
{
    /* System entry prompt */
    std::cout << "+-----+" << std::endl;
    std::cout << "|          命题逻辑推理          |" << std::endl;
    std::cout << "| Propositional Logic Reasoning |" << std::endl;
    std::cout << "+-----+" << std::endl <<
std::endl;

    /* Problem description */
    std::cout << ">>> 问题描述" << std::endl;
    std::cout << "    [命题 1] 营业员 A 或 B 偷了手表" << std::endl;
    std::cout << "    [命题 2] 若 A 作案，则作案不在营业时间" << std::endl;
    std::cout << "    [命题 3] 若 B 提供的证据正确，则货柜未上锁" <<
std::endl;
    std::cout << "    [命题 4] 若 B 提供的证据不正确，则作案发生在营业时间"
<< std::endl;
    std::cout << "    [命题 5] 货柜上了锁" << std::endl << std::endl;
```

```

std::cout << ">>> 命题变元表示" << std::endl;
std::cout << "    [命题 A] 营业员 A 偷了手表" << std::endl;
std::cout << "    [命题 B] 营业员 B 偷了手表" << std::endl;
std::cout << "    [命题 C] 作案不在营业时间" << std::endl;
std::cout << "    [命题 D] B 提供的证据正确" << std::endl;
std::cout << "    [命题 E] 货柜未上锁" << std::endl << std::endl;

/* Propositional logic reasoning */
for (int i = 0; i < pow(2, 5); i++) {
    bool A = i & 0b1;
    bool B = i & 0b10;
    bool C = i & 0b100;
    bool D = i & 0b1000;
    bool E = i & 0b10000;
    if ((A || B) && (!A || C) && (!D || E) && (D || !C) && !E) {
        std::cout << ">>> 命题逻辑推理结果: A = " << (A ? "True" :
"False") << ", B = " << (B ? "True" : "False") << std::endl <<
std::endl;
        std::cout << ">>> 所以营业员 " << (A ? "A" : "") << (B ?
"B" : "") << " 偷了手表" << std::endl;
    }
}

/* Wait for enter to quit */
std::cout << std::endl << "Press Enter to Quit" << std::endl;
while (_getch() != '\r')
    continue;

/* Program ends */
return 0;
}

```