《离散数学》课程实验报告 2 命题逻辑推理

2253893 苗君文 软件工程

1. 题目简介

1.1. 背景与目的

命题逻辑推理涉及使用逻辑规则和真值表，以确定复合命题在给定条件下的真假值。通过推理，我们可以得出关于不同命题之间关系的结论，从而解决问题或证明论断。在计算机科学、人工智能等领域，命题逻辑是一种重要的推理工具，用于表达和分析复杂的逻辑关系。

本项目旨在加深对命题逻辑推理方法的理解，学会用命题逻辑推理的方法解决逻辑推理问题。

1.2. 问题描述

根据下面命题，用命题逻辑推理方法确定谁是作案者，并给出推理过程，C++语言源代码及演示界面。

（1）营业员 A 或 B 偷了手表；

（2）若 A 作案，则作案不在营业时间；

（3）若 B 提供的证据正确，则货柜未上锁；

（4）若 B 提供的证据不正确，则作案发生在营业时间；

（5）货柜上了锁。

1.3. 程序输入输出

1.3.1. 输入

本项目由于所有的取值都是根据问题本身得来，因此不需要额外的输出。而程序中有一系列真值，代表命题中的变量的真值，这些变量分别是：

A：表示营业员 A 是否偷了手表，取值为 0 或 1。

B：表示营业员 B 是否偷了手表，取值为 0 或 1。

C：表示作案是否发生在营业时间，取值为 0 或 1。

D：表示 B 提供的证据是否正确，取值为 0 或 1。

E：表示货柜是否上了锁，取值为 0 或 1。

1.3.2. 输出

程序的输出包括两个部分：

（1）程序的命令行输出：输出符合所有命题的变量组合，其中包括营业员A或B是否偷了手表。

（2）演示界面：通过在控制台打印相关信息，展示逻辑推理的过程和最终的结论。

2. 解题思路

2.1. 明确命题逻辑推理相关概念

2.1.1. 命题

命题是判断结果唯一的陈述句，陈述句中的悖论以及判断结果不唯一确定的也不是命题。命题分为简单命题和复合命题。简单命题是简单陈述句构成的命题，复合命题则是由简单命题通过联结词联结而成的陈述句。

2.1.2. 联结词

（1）非：表示为 ¬，它对命题的真假进行否定。例如，¬p 表示 ″非 p″，如果 p 为真，则¬p 为假，反之亦然。

（2）合取：表示为∧，它对两个命题进行″与″操作。例如，p∧q 表示 ″p 且 q″，只有当 p 和 q 都为真时， p∧q 才为真。

（3）析取： 表示为∨，它对两个命题进行″或″操作。例如， p∨q 表示 ″p 或 q″，只要 p 或 q 有一个为真，p∨q 就为真。p∨q 为假当且仅当 p 与 q 同时为假。

（4）蕴含：表示为→，它描述了一种″如果...那么...″的关系。p→q 表示 ″如果 p，则 q″，当 p 为假或者 q 为真时，p→q 为真；p→q 为假当且仅当 p 为真且 q 为假。

（5）等价：表示为↔，它描述了两个命题之间的等价关系。例如，p↔q 表示 ″p 当且仅当 q″，p↔q 为真当且仅当 p 与 q 同时为真或同时为假。

2.1.3. 真值表

真值表是列出了命题的所有可能组合及其结果的表格。通过真值表，我们可以确定复合命题在不同情况下的真假值。命题公式在所有可能的赋值下的取值的列表含 n 个变项的公式有 2^n 个赋值。

2.2. 实现思路

本项目采用类似于真值表的方法，类似于枚举的方式，将使用嵌套的 for 循环遍历所有可能的变量组合。对于每个组合，我们将检查是否满足所有给定的命题。如果满足，将输出这个变量组合。具体实现步骤如下：

（1）符号化题目中的命题得到：

A: 营业员 A 偷了手表

B: 营业员 B 偷了手表

C: 作案不在营业时间

D: B 提供的证据正确

E: 货柜未上锁

（2）将它们作为条件，得出一个复合命题。

（3）将复合命题中要用到的联结词定义成 C++语言中的函数，用变量表示相应的命题变元，将复合命题写成一个函数表达式，符号化的命题为：

(A || B) && (!A || C) && (!D || E) && (D || !C) && !E

（4）函数表达式中的变量赋初值为 1。如果函数表达式的值为 1，则结论有效，A 偷了手表，否则是 B 偷了手表。

3. 所用数据结构

在这个项目中，并没有直接使用复杂的数据结构。主要的数据结构是基本的变量，如 int，用于表示命题中的不同状态。程序通过嵌套的 for 循环遍历所有可能的变量组合，并通过 if 语句检查是否满足所有命题。

虽然这里没有使用复杂的数据结构，但在实际的软件开发中，如果问题的复杂性增加，可能会考虑使用更高级的数据结构，例如图、树、队列等，以更有效地组织和处理数据。在这个简单的逻辑问题中，基本的控制结构足以完成任务。

4. 核心算法

这个项目主要使用了命题逻辑的推理方法来确定作案者。核心算法是通过嵌套的 for 循环遍历所有可能的变量组合，并通过 if 语句对每个组合进行条件判

断，找到符合所有命题的解，从而确定作案者。

以下是核心算法的步骤：

（1）定义变量：为了表示不同的情况，定义了五个变量 A、B、C、D、E，它们分别表示营业员 A 是否偷了手表、营业员 B 是否偷了手表、作案是否在营业时间、B 提供的证据是否正确、货柜是否上锁。

（2）使用 for 循环遍历所有可能的组合：嵌套了五个 for 循环，分别用于遍历 A、B、C、D、E 的取值，取值范围是 0（假）或 1（真）。这样，通过这五个变量的组合，可以穷举所有可能的情况。

（3）应用命题逻辑： 利用逻辑命题的条件进行判断。在 if 语句中，使用了命题逻辑的条件来筛选符合问题描述的情况。例如，(A || B) 表示 A 或 B 偷了手表，(!A || C) 表示如果 A 偷了手表，则作案不在营业时间。通过这些条件的组合，筛选出符合所有命题的情况。

（4）输出结果： 如果找到了满足所有命题的组合，通过 cout 语句输出相应的信息，指示哪位营业员偷了手表。

这种方法是一种穷举法，通过遍历所有可能的情况来寻找符合特定条件的解。在这个简单的逻辑问题中，由于变量数目较少，这种穷举法是可行的。在实际的问题中，如果变量数目较大，可能需要考虑其他更高效的算法。

```
int main()
{
    /* A:营业员A偷了手表
       B:营业员B偷了手表
       C:作案不在营业时间
       D:B提供的证据正确
       E:货柜未上锁        */
    int A, B, C, D, E;
    for (A = 0;A <= 1;A++)
        for (B = 0;B <= 1;B++)
            for (C = 0;C <= 1;C++)
                for (D = 0;D <= 1;D++)
                    for (E = 0;E <= 1;E++)
                        if ((A || B) && (!A || C) && (!D || E) && (D || !C) && !E) {
                            cout << "A=" << A << ", B=" << B << endl;
                            if (A) cout << "A偷了手表。" << endl;
                            if (B) cout << "B偷了手表。" << endl;
                        }
    return 0;
}
```

5. 心得体会

这个项目通过命题逻辑推理方法解决了一起偷窃案中作案者的问题，让我对命题逻辑这一章有了更为深刻的理解。通过编写程序，能够系统地遍历所有可能的情况，找到符合所有条件的变量组合。这展示了在复杂逻辑问题中使用计算机进行系统分析和解决问题的能力。通过简单的演示界面，可以清晰地展示逻辑推理的过程，使问题的解决变得更加可视化。

6. 测试结果