**离散数学课程设计**

**项目说明文档**

**命题逻辑推理**

作 者 姓 名： 苏家铭

学 号： 2151299

指 导 教 师： 唐剑锋

学院、 专业： 软件学院 软件工程

同济大学

**Tongji University**



**目录**

[1 项目分析 1](#_Toc4253)

[1.1 项目背景 1](#_Toc12641)

[1.2项目要求 1](#_Toc11232)

[1.2.1 功能要求 1](#_Toc26032)

[1.2.3 输出要求 1](#_Toc15693)

[2 项目设计及实现 2](#_Toc25723)

[2.3 项目算法 2](#_Toc27979)

[2.3.1 实现思路 2](#_Toc32382)

[2.3.2 代码实现 2](#_Toc5102)

[3 项目测试 3](#_Toc21683)

[4 算法性能分析 3](#_Toc14999)

[4.1 正确性 3](#_Toc19968)

[4.2 可使用性 3](#_Toc19318)

[4.3 可读性 3](#_Toc22323)

[4.4 健壮性 4](#_Toc26133)

[5 实验感想 4](#_Toc10193)

# 1 项目分析

## 项目背景

推理是从前提推出结论的思维过程，前提是指已知的命题公式，结论是指从前提出发应用推理规则推出的命题公式。前提可以是多个，由前提A1,A2,...Ak可以推出结论B。

在现实生活中，常常遇到需要通过一系列已知的前提推出一个未知的结论的过程，而往往前提又有很多个。假若仅通过思维上的枚举很难得出正确的结果，在信息时代我们可以利用程序枚举，能在一个很短的时间内求解出前提的真值。

研究命题逻辑推理对于个人思维的提升具有非常大的好处，对于现实问题的分析也具有重要的社会效益。

## 1.2项目要求

### 1.2.1 功能要求

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。

### 1.2.3 **输出要求**

能使结论成立的命题的真值。

# 2 **项目设计及实现**

## 2.3 **项目算法**

### 2.3.1 实现思路

1. 自定义符号化以上5个命题，把它们作为推理的前提条件，合并为一个复合命题。
2. 把复合命题中要使用到的连接词定义成C++高级语言程序中的二元逻辑连接词，例如‘&&’、‘||’等，用变量表示命题变元，从而将复合命题表示成一个程序能识别的函数表达式。
3. 利用穷举法，枚举出每种可能的情况，若有一种情况能使得函数表达式值为真(true)，则该表达式中的变量对应的真值就是所求命题的真值结果。

本例题的命题用命题变元表示如下：

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

有题意可知上面的命题可符号化为：

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

要求找到满足上面式子的变元A，B的指派便是结果。

### 2.3.2 代**码实**现

#include <iostream>

using namespace std;

int main()

{

int A, B, C, D, E;

//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

bool success = false;//健壮性，有无结果

for (A = 0; A <= 1; A++)//营业员A偷了手表与否

for (B = 0; B <= 1; B++)//营业员B偷了手表与否

for (C = 0; C <= 1; C++)//C : 作案不在营业时间与否

for (D = 0; D <= 1; D++)//D : B提供的证据正确与否

for (E = 0; E <= 1; E++)//E : 货柜未上锁与否

if ((A || B) && (!A || C) && (!D || E) && (D || !C) && !E){//若满足该式子

cout << "A=" << A << ",B=" << B << endl;//输出结果

success = true;//找得到指派

}

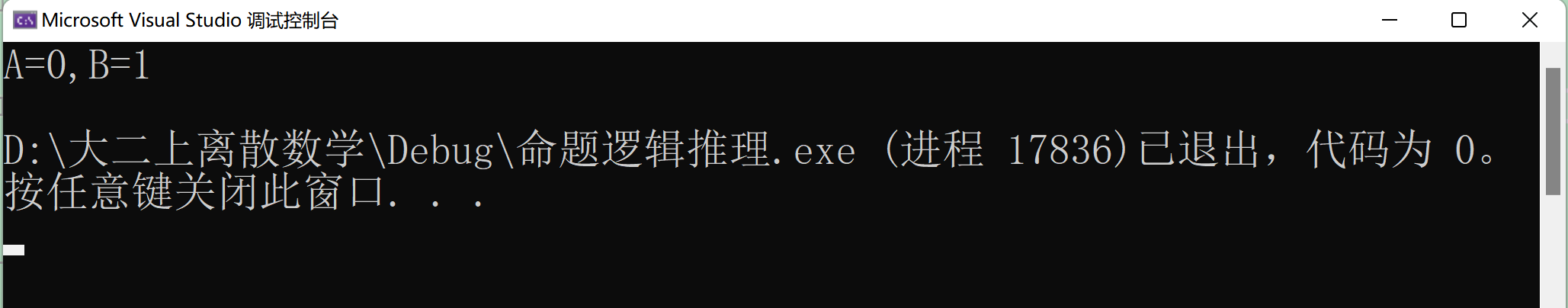
if(success==false)

cout << "没有相应真值指派" << endl;

return 0;

}

# 3 项目测试



# 4 算法性能分析

## 4.1 正确性

本算法能正确地执行预定的功能和性能要求，由以上版块证明本项目与例题的情况比较后满足正确性。

## 4.2 可使用性

本算法可以很方便地使用，求命题逻辑推理功能只需更换命题个数与循环次数便可在另一个情况下使用，并且该算法有良好的界面和完备的用户文档。没有使用公用变量或全局变量。

## 4.3 可读性

本算法逻辑清晰、简单、且结构化，所有命名与函数名都具有实际含义，让人见名知义。且算法中包含了大量注释，简要说明了算法功能、输入与输出参数的使用规则、重要数据的作用、算法中各程序段完成的功能。

## 4.4 健壮性

本算法对于边界条件，诸如：没有找到相应命题指派的情况有错误提示。

# 5 实验感想

命题逻辑推理是离散数学中非常重要的一个知识，不仅考验我的思维严谨度，也考验我对离散数学推理概念的理解。利用程序设计的方法得到一个复合命题的真值指派能让我对穷举法的思考又进一步。虽然在算法学习的过程中穷举法是最简单但又最耗费时间的方法，但是对于一个命题变元而言，它的真值不是0就是1，只有两种可能，每一层循环也就只有两种可能的分支，所以整个求解能使得结论成立的过程无异于一颗满二叉树的深度搜索。若有n个命题变元，则会有2n种不同的情况，对于n较小的情况下都是可行的一种方法。

虽然本项目并不需要耗费太久的时间，但我也深刻理解了逻辑推理的过程，这就像绘制一个真值表，不断地“试错”，直到最后得出正确的结果。这种方法不像构造证明法，也没有利用绘制真值表、计算主析取范式的方式，而是利用了一种最可简单的程序遍历方式得到能使结论成立的所有真值指派。对于正在软件工程学习的学生来说，具有非常重要而深远的意义。