

## 《离散数学课程项目文档》 ——命题逻辑推理

作者姓名：\_\_\_\_\_胡峻玮\_\_\_\_\_

学        号：\_\_\_\_\_2153393\_\_\_\_\_

指导教师：\_\_\_\_\_唐剑锋\_\_\_\_\_

学院、专业：\_\_\_\_\_软件学院    软件工程\_\_\_\_\_



## 目 录

1 项目分析.....	1
1.1 项目背景 .....	1
1.2 项目要求 .....	1
2 项目设计.....	1
2.1 数据结构设计.....	1
2.2 算法设计.....	2
2.2.1 算法思路.....	2
2.2.2 性能评估.....	2
2.2.3 流程图表示.....	2
2.2.4 代码实现.....	3
3 项目测试.....	4
4 心得体会.....	5

## 1. 项目分析

### 1.1 项目背景

在日常生活中，我们常常需要运用命题逻辑推理的技巧，特别是在解决问题或破案的过程中。在这些情境中，通常会有大量的信息可供我们使用，但这些信息可能包含真实和虚假的元素，而且它们之间可能会相互关联，因此我们需要进行深思熟虑的推理，以辨别事实和虚构，最终找到真相。

### 1.2 项目要求

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。

## 2. 项目设计

### 2.1 数据结构设计

由项目分析可以得出，该项目需要完成一次推理的模拟。在推理过程中，我们采用 A, B, C, D, E 分别代表五个命题：

- 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: 营业员 B 提供的证据正确

E: 货柜未上锁

## 2.2 算法设计

### 2.2.1 算法思路

推理对于人来说可以使用复杂但快捷的推理方法，可对于计算机来说无法进行如此灵活的运算，而最合适的方法就是考虑所有的情况，再一一比对哪一种情况是对的。

拿本题中的 5 个命题来说，共包括了 5 个原子命题。抛开其内在联系不说，组合共  $2^5=32$  种情况，又判别组合是否符合情况的表达式为：

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

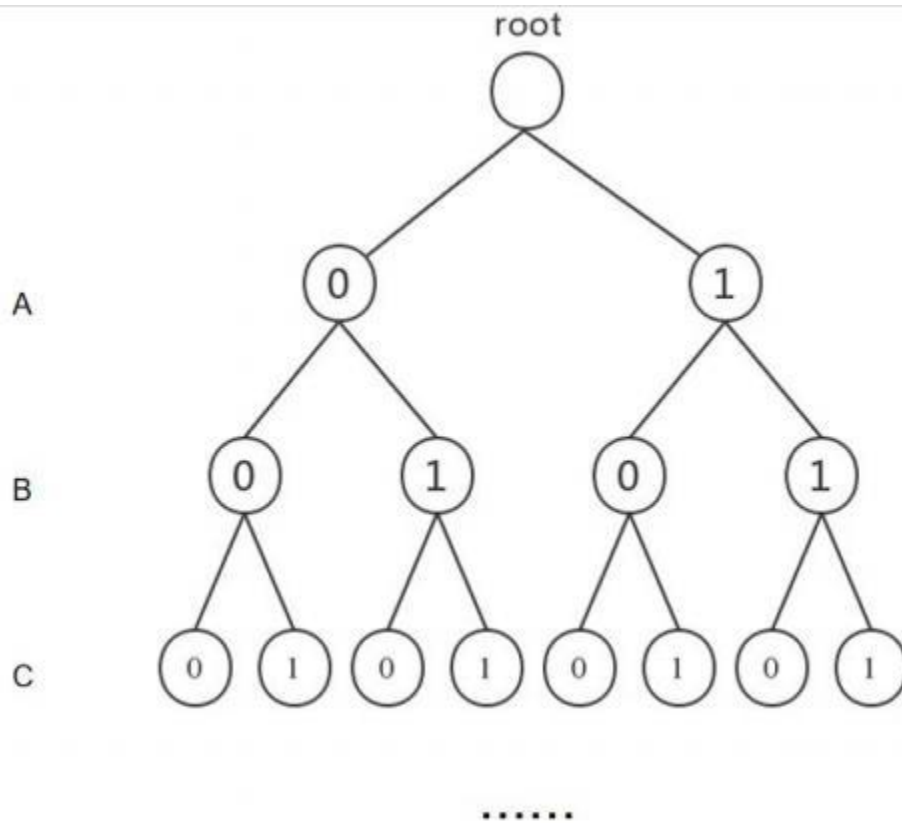
故可以把每一种情况转换为一个二进制序列，代表真假的一种组合（例如第 10 种转换为 01010，代表 5 个原子命题情况为“假真假真假”，然后代入到表达式判断。每个命题均有两种可能，将每个命题的不同取值进行组合，若结果为真，则得出一种推理结果，将其输出。该过程利用五个循环来执行，让 A、B、C、D、E 分别取到 0 和 1 进行判断，最后输出使上述表达式为真的 A B 的取值

### 2.2.2 性能评估

最内层语句要执行  $2^n$ ，则时间复杂度为  $O(2^n)$ 。

### 2.2.3 流程图表示

由于流程过长，故只展现部分，对 A B C D E 分别取 0 1 的情况进行组合得到共 32 种不同的组合，但有且仅有一组取值使表达式为真，则输出该组取值的 A B 值



## 2.2.4 代码实现

```
#include <iostream>
using namespace std;

int main()
{
    cout << "命题如下: " << endl;
    cout << "(1) 营业员 A 或 B 偷了手表" << endl;
    cout << "(2) 若 A 作案, 则作案不在营业时间" << endl;
    cout << "(3) 若 B 提供的证据正确, 则货柜未上锁" << endl;
    cout << "(4) 若 B 提供的证据不正确, 则作案发生在营业时间" << endl;
    cout << "(5) 货柜上了锁" << endl;
    cout << endl;
    cout << "用命题变元表示如下: " << endl;
    cout << "A : 营业员 A 偷了手表" << endl;
    cout << "B : 营业员 B 偷了手表" << endl;
    cout << "C : 作案不在营业时间" << endl;
    cout << "D : B 提供的证据正确" << endl;
    cout << "E : 货柜未上锁" << endl;
```

```

cout << endl;

cout << "命题与命题变元的对应表示为: " << endl;

cout << " (1) 营业员 A 或 B 偷了手表                ->  A  | B " << endl;
cout << " (2) 若 A 作案, 则作案不在营业时间          ->  A  -> C " << endl;
cout << " (3) 若 B 提供的证据正确, 则货柜未上锁        ->  D  -> E " << endl;
cout << " (4) 若 B 提供的证据不正确, 则作案发生在营业时间 ->  !D->!C " << endl;
cout << " (5) 货柜上了锁                                ->  !E " << endl;

for (int A = 0; A <= 1; A++)
    for (int B = 0; B <= 1; B++)
        for (int C = 0; C <= 1; C++)
            for (int D = 0; D <= 1; D++)
                for (int E = 0; E <= 1; E++)
                    if ((A || B) && (!A || C) && (!D || E) && (D || !C) && !E)
                    {
                        cout << endl << "A=" << A << ", B=" << B << "时, 满足以上全部条件" << endl;
                        cout << endl;

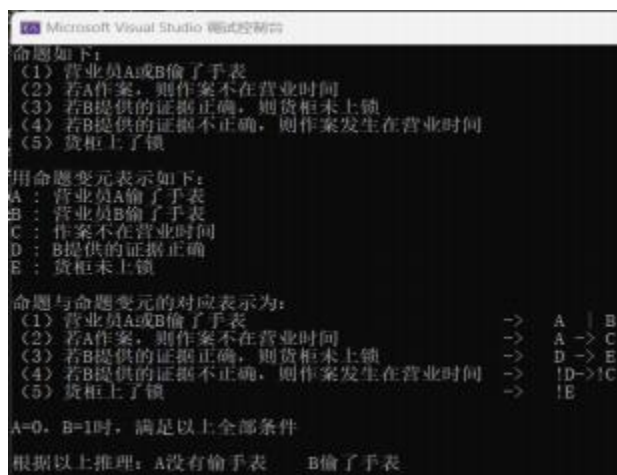
                        cout << "根据以上推理: ";
                        A ? cout << "A 偷了手表" : cout << "A 没有偷手表";
                        cout << "      ";
                        B ? cout << "B 偷了手表" : cout << "B 没有偷手表";
                        cout << endl;
                    }

return 0;
}

```

### 3. 项目测试

由运行结果可知, 是营业员 B 偷了手表, A 没偷, 且只有这一种情况。



## 4. 心得体会

通过完成本次作业，我更加深入地理解了命题逻辑推理的方法。这过程中，我学会了如何利用计算机来进行逻辑推理，这对于解决复杂问题非常有用。此外，我也掌握了将任务分解成各个模块来实现的分治思想，这有助于提高问题的解决效率。

通过编写代码，我能够将抽象的逻辑问题具体化，并通过计算机的计算能力来验证我的推理。这种实践经验使我更加熟悉了逻辑运算符和条件语句的使用，同时也提高了我的编程技能。

总之，本次作业不仅增强了我对命题逻辑推理方法的理解，还加强了 my 计算机编程能力。我认识到分治思想在解决复杂问题时的重要性，并将继续努力提升自己在逻辑推理和编程领域的能力。