**《离散数学》课程实验报告**

# 《离散数学》课程实验报告2

命题逻辑推理

作 者 姓 名： 毛凌骏

学 号： 2053058

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

1. **题目简介**

**1.1实验内容：**

根据下面命题，用命题逻辑推理方法确定谁是作案者

（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实验环境：**

采用C＋＋编程语言， Visual Studio Code 2019实验环境实现。

1. **解题思路**

**2.1 命题逻辑与命题变元**

命题逻辑是用数学方法研究人的思维形式和规律，通过建立一套表意符号体系对事物进行抽象并推演，从而研究前提和结论间的形式关系的学科，其根据研究对象和方法的不同分为形式逻辑、辩证逻辑、数理逻辑。而命题是指结果或为真或为假，即存在唯一的真值，而不是两者同时成立的陈述句，其中不包含逻辑连接词的命题是简单命题。通过命题逻辑的推演我们可以求解很多现实生活中的问题，比如本题中对谁是作案者的推断，借助命题逻辑公式的形式可将实际问题符号化，之后通过遍历或数学演算求解即可。

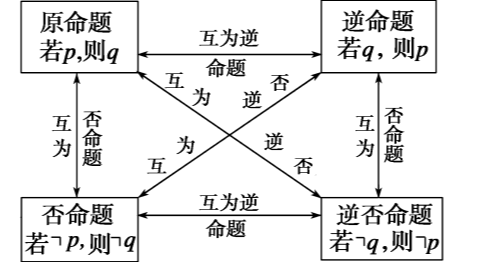


图 1 命题与命题之间的关系

**2.2 命题变元的符号化**

符号化上面的命题，将它们作为条件，得出一个复合命题。

**2.3 命题的符号化**

用命题变元表示：

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

则上面的命题可符号化为：(A||B) && (!A||C) && (!D||E) && (D||!C) && !E

**2.4 命题公式的求解**

真值表是表征逻辑事件输入和输出之间全部可能状态的表格，通常以1表示真，0 表示假。命题公式的取值由组成命题公式的命题变元的取值和命题联结词决定，命题联结词的真值表给出了真假值的算法。真值表是在逻辑中使用的一类数学表，用来确定一个表达式是否为真或有效，对于一个命题公式，我们可以通过遍历的方式对其中的命题变元依次取T与F的值，求出该命题公式对应的真值表，真值表中真值为F所对应的命题变元的取值即为该命题公式的解。

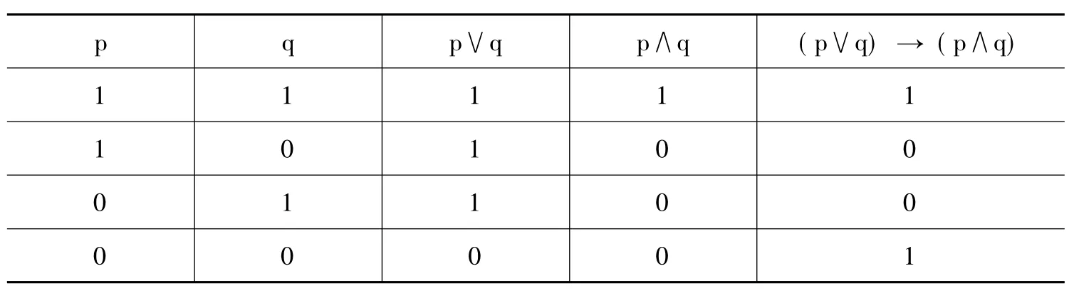


图 2 真值表示例

当然对于较为复杂的命题公式，采用对其中出现的所有命题变元依次赋值的方式求解可能会使程序的复杂度过高。假设一个命题公式中包含n个命题变元，则整个程序的时间复杂度为，循环层数太多，求解效率过低且程序可读性不高。为此，我们可以通过一些人为的逻辑演算简化程序运行过程。常见的简化方式有逻辑短路。

C语言中，表示条件时用关系表达式，即由关系运算符 >、>=、<、<=、==、!=六个运算符之一连接起来的表达式。当表示复杂条件时，可以将多个关系表达式用逻辑运算符连接，构成逻辑表达式。C语言中的逻辑运算符有3个：&& 用于连接两个条件，当这两个条件同时成立时，整个条件成立。||用于连接两个条件，这两个条件只要有一个成立，整个条件就成立。!只在其后连接一个表达式，取与这个表达式结果相对的逻辑值，即真变假，假变真。

在&&的使用中，如a > 0 && a<= 100，a的值如果等于 -5，这时条件的前一部分就不成立，对于&&来说，不管后面的条件是否成立，整个条件都不会成立，或者说&&后面的条件执不执行都不会改变整个表达式的取值，这样后半部分实际也就没必要执行了。在 C语言中，这种一部分代码被跳过、不执行的情况就叫逻辑短路。&&可以构成逻辑短路，||也可以，||的功能是两个条件只要有一个成立，整个条件就成立。通过逻辑短路的设计，我们可以在命题公式求解的过程中减少计算机尝试的次数，即算法“剪枝”的思想。

1. **核心算法**

**3.1 命题符号化的实现**

已知下列5个命题：

（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。

分析上述命题后，我们可以提炼出其中的原子命题，并用命题变元表示：

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

得到原子命题之后，我们就可以通过联结词用原子命题表示上述命题：

命题1：营业员A或B偷了手表

命题1属于相容的选言命题，指几个可能的事物情况中至少有一个存在并且可以同时存在的选言命题，这里我们使用联结词来表示。故命题1可以表示为，用计算机语言可表示为A||B。

命题2：若A作案，则作案不在营业时间

命题2属于蕴含式，表示如果...则...。其中我们将如果F,则...的情况全部视为真，这在离散数学中称为善意断言。因此，命题2可表示为，借助蕴含式的运算性质可简化为，用计算机语言可表示为!A||C。

命题3：若B提供的证据正确，则货柜未上锁

与命题2同理，可表示为，用计算机语言可表示为!D||E。

命题4：若B提供的证据不正确，则作案发生在营业时间

与命题2同理，但多了一个否定的复合，可表示为!，用计算机语言可表示为D||!C。

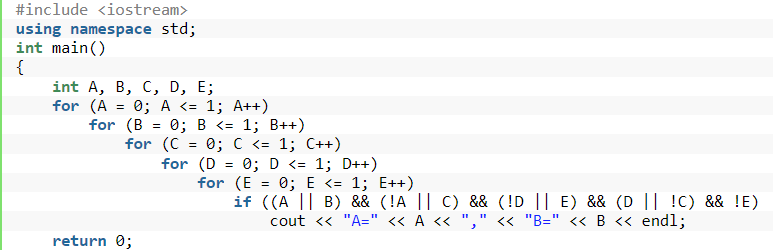
命题5：货柜上了锁

命题5是命题E的负命题。负命题就是通过否定某个命题所得到的命题，又称命题的否定。设原命题为P，则该命题的负命题为“并非P”或者“非P”。故命题5可表示为!E，用计算机语言表示也为!E。

故本实验的命题可符号化为：(A||B) && (!A||C) && (!D||E) && (D||!C) && !E

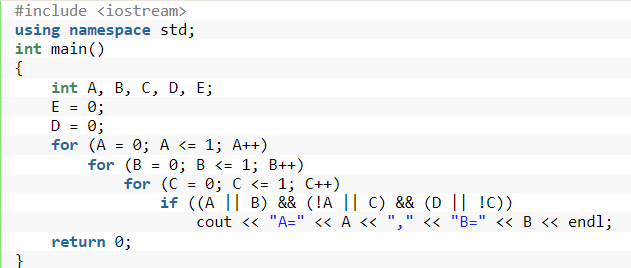
**3.2 命题公式求解的实现**

由于本实验中命题变元的个数较少，共A、B、C、D、E五个命题变元，故命题公式的求解可采用依次遍历命题变元判断真值的方式。



1. **算法优化**

为简化程序的复杂度，我们可以采用人为逻辑推理的方式进行一部分预处理。观察到(A||B) && (!A||C) && (!D||E) && (D||!C) && !E中每两项之间都是由联结词相连接的，故若使整体为真，其中每一项都得为真。根据!E项可知E的取值是唯一确定的必须为0，而根据 !D||E一项可知在E为0的前提下D必须为0，才能使整体为真。这样根据逻辑运算中短路的思想我们直接确定了两项命题变元的真值，减少了程序的两重循环。



1. **心得体会**

命题推理是指由若干命题得出一个命题的思维过程。其中能提供100%证据支持度的推理称为[必然性推理](https://baike.baidu.com/item/%E5%BF%85%E7%84%B6%E6%80%A7%E6%8E%A8%E7%90%86/4488937?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)，只能提供某种小于100%证据支持度的推理称为[或然性推理](https://baike.baidu.com/item/%E6%88%96%E7%84%B6%E6%80%A7%E6%8E%A8%E7%90%86/4488952?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/_blank)。一般地，演绎推理是必然性推理，归纳推理、类比推理等是或然性推理。

在本实验中我们尝试了将一个实际问题化归成了命题公式求解的问题，再次熟悉了C语言本身支持的与&&、或||、非!三种逻辑运算符，学会了如何将一个复杂的问题拆分成若干简单原子命题，并通过各项联结词将其连接。对于命题的公式求解，我们可采用依次遍历命题变元判断真值的方式，而如果命题变元数量过多，也可采用人为逻辑推理的方式进行一部分预处理，以降低算法的复杂度。