《离散数学》课程实验报告

2-命题逻辑推理

一、题目简介

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++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。

二、解题思路与算法

考虑用命题逻辑推理的方法解决逻辑推理问题。

①首先符号化上面的命题，将它们作为条件，得出一个复合命题。再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。

本项目使用的命题变元为

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。

其中 A||B 表示 A 或 B 作案；!A || C 表示 A 没有作案或作案不在营业时间；!D || E 表示 B 提供的证据不正确或货柜未上锁；D || !C 表示 B 提供的证据正确或作案在营业时间；!E 表示货柜上锁了。

在命题符号化定义变量时，我原本是想要将变量都定为 bool 型，因为只有 0 和 1 两种情况，还可以节省内存。但考虑到要用循环嵌套，在写循环判定语句时变量小于等于 1 的条件有可能不安全，于是想到另一种方法：

再定义一个 translate 函数，用于将十进制数转化为二进制数分位数存放进数组，这样可以减少循环嵌套所用的时间。

②接下来用循环嵌套模拟 ABCDE 分别为 0 或 1 时命题表达式的值。当表达式的值为 1 时，结论有效，此时输出对应的 A 和 B 的值，若 A 为 1，则 A 偷了手表，否则是 B 偷了手表。

例题代码给出时，if 语句执行后意味着找到偷手表的人，应当终止循环的继续，保证代码的运行高效性。于是我在 if 语句中加入 return 0;终止 main 函数，返回函数值。

③后续修改完善。为了使代码看上去不那么冗长、更美观些，我将变量的定义最终都放入各自的 for 循环初始化语句中。其次为了将输出结果更详细易懂，我在 if 语句中新增了 cout << "So " << (A ? 'A' : 'B') << " is the thief who stole the watch." << endl;，使得结果更明确地指出最终答案推出的结果。

三、代码与运行结果

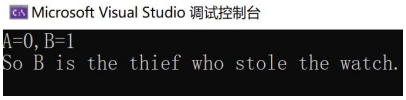
```
#include <iostream>

using namespace std;                                    bool translate(int dec,int bin[5]) {
```

```
	//参数为十进制的数和存放转换后二进制的数组
	if (dec >31)
		return false;//当十进制数大于2^5-1时，对应二进制数超过5位，与题目对应命题变量不符
	int i = 4;//用以标记数组下标进行存放
	while (dec / 2 > 0) {
		bin[i--] = dec % 2;
		dec = dec / 2;
	}
	dec = bin[i];//最后一位数必为1，存入数组
	return true;
}

int main()
{
	int bin[5] = { 0 };
	/*
	bin[0]对应营业员A偷了手表
	bin[1]对应营业员B偷了手表
	bin[2]作案不在营业时间
	bin[3]B提供的证据正确
	bin[4]货柜未上锁
	*/
	for (int all_cases = 0; all_cases < 32; all_cases++) {//达到2^5-1后结束循环
		translate(all_cases, bin);
		if ((bin[0] || bin[1]) && (!bin[0] || bin[2]) && (!bin[3] || bin[4]) && (bin[3] || !bin[2]) && !bin[4]) {
			//A||B：A或B作案
			//!A || C：A没有作案或作案不在营业时间
			//!D || E：B提供的证据不正确或货柜未上锁
			//D || !C：B提供的证据正确或作案在营业时间
			//! E：货柜上锁了
			cout << "A=" << bin[0] << ",B=" << bin[1] << endl;
			cout << "So " << (bin[0] ? 'A' : 'B') << " is the thief who stole the watch." << endl;
			return 0;
		}
	}
	return 0;
}
```

运行结果：



四、体会与心得

本题目看着不算难，因而也是分数最少的一项，但认真操作起来还是有一些思维逻辑的力度在其中的，特别是将连接词转换成C语言的符号。

例如“若B提供的证据正确，则货柜未上锁”这一句，在命题逻辑语言里是B→E，这里需要进行蕴含等值式的演算，将B→E等值为!B∪E，再转换为C语言中的!B||E。所以将题目中的5个命题都进行转换的过程中需要特别仔细，一旦一个符号有错，那么“偷手表的小偷”就可能找错了。

最后，该题目让我对现实案件、命题逻辑和程序语言的分段式了解有了新的认知：这三个是可以相互联系并转化的。现实问题可以经过命题逻辑作为媒介，转换成程序语言，通过系统更快速地获得答案，不失为人们的一大高效工具。