



软件设计和开发

Mini Project  
设计与测试报告  
2016 Spring

# Table of Content

<b>1 Overview</b>	<b>2</b>
1.1 问题描述 . . . . .	2
1.2 解题概况 . . . . .	2
<b>2 主要思路</b>	<b>3</b>
2.1 expr_to_truthtable . . . . .	3
2.2 truthtable_to_expr . . . . .	3
<b>3 程序结构</b>	<b>4</b>
<b>4 核心思路</b>	<b>5</b>
<b>5 测试</b>	<b>6</b>
<b>6 总结与体会</b>	<b>7</b>

# 1 Overview

## 1.1 问题描述

- 编写一个函数，计算逻辑表达式的真值表
  - 逻辑表达式最多有 8 个输入项，分别为  $A, B, C, D, E, F, G, H$ ；
  - 支持的逻辑运算符按优先级从高到低分别为  $\sim$ （非）、 $\&$ （与）、 $\wedge$ （异或）、 $|$ （或）；
- 编写一个函数，根据真值表计算逻辑表达式，以字符串格式输出
  - 使用 Quine-McCluskey 算法对逻辑表达式进行化简。
- 函数接口：
  - `std::string expr_to_truthtable(int n, const std::string& expr);`  
其中  $n$  是变量个数， $expr$  是逻辑表达式，返回真值表。
  - `std::string truthtable_to_expr(const std::string& truth_table);`  
其中  $truth\_table$  是真值表，返回逻辑表达式。

若参数无效，函数抛出异常。

- 要求
  - 使用第 5 讲的测试框架对两个函数进行测试；提交源代码与设计文档。
  - 文档内容包括：设计思路、数据结构与算法、重要的类与函数的说明、测试用例的设计等等，也可以写完成 Project 的心得体会、经验与教训等。

## 1.2 解题概况

最终程序各部分及其作用如下表

Documents	Functions
constants.h	常量的定义
simple_test.h	参与测试
ChartToExpression.h	将真值表转换为逻辑表达式
ExpressionToChart.h	将逻辑表达式转换为真值表
implication	质蕴涵项的实现

下面就各部分的解题思路及实现进行详细讨论

## 2 主要思路

本次 Project 分为两部分，下面就这两部分分别进行讨论：

### 2.1 `expr_to_truthtable`

首先，通过将题目抽象，这一部分是一个经典的表达式求值的问题，其主要问题如下：

利用算符优先关系，实现对给定混合运算表达式的求值。

这是个栈的经典应用问题，常见解法是将中缀表达式（即输入的表达式）转化为后缀表达式后求解。时间复杂度为  $O(n)$ ，其中  $n$  为变量的个数

具体的算法会在 `main_algorithm` 模块详细介绍。

表达式求值是一个非常经典的问题，已经有了许多成熟的解法。甚至在 Python 中，一条 `evaluate` 语句就可以完成这样的工作。但是对于含字母的表达式，至今没有太好的解决办法。

但是考虑到本次作业的特殊性，一共只有最多 8 个变量，同时是逻辑计算而不是算术计算，这无疑极大的降低了难度。对于普通的算术含参表达式，我们只能多次取不同的大素数代入变量，还不能保证正确性。但是在本题的布尔代数式中，一共只有  $2^8 = 256$  种取值方式，是完全可以接受的。于是，我们可以先遍历所有变量的可能取值，代入表达式中，从而计算出真值表中对应的值。

因此，总的时间复杂度为  $O(n * 2^n)$ ，由于  $n \leq 8$ ，复杂度不会超过  $8 * 2^8 = 2048$ ，是完全可以接受的。

### 2.2 `truthtable_to_expr`

对于将真值表转化为表达式的问题，我们在数字逻辑电路中曾学习过使用卡诺图的方法，非常直观，同时在对于维数较低时也易于化简。

但是卡诺图的解题过程较为繁琐，同时对于高维的情况扩展性较差。在  $n \geq 6$  时，就比较难画出合适的卡诺图了。因此，我们选用 Quine-McCluskey 算法。它虽然较为抽象，但是非常模式化，对高维的扩展性很好。

至于算法的时间复杂度方面，由于化简表达式至今还是 NP-hard 问题，因此只存在指数级解法，即问题的时间复杂度为  $O(2^n)$

同时，题目要求将表达式化到最简形式，因此在 Quine-McCluskey 算法后我们必须进行化简。这里采用 petrick 化简算法，时间复杂度同样为  $O(2^n)$

具体的算法会在 `main_algorithm` 模块详细介绍。

### 2.3 异常处理

我还没做呢

### 3 程序结构

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 4 核心思路

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 5 测试

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 6 总结与体会

很惭愧，就做了一点微小的工作，谢谢大家