

QM 算法上机作业——接口

以课件上的 QM 例子为例

$$f(A, B, C, D) = \sum m(4, 8, 10, 11, 12, 15) + d(9, 14)$$

我们要实现的函数以两个数组 m 、 d 为输入，输出一个最简表达式 $BC'D'+AC+AB'$ 或者 $BC'D'+AC+AD'$ 。

附件中的 QMTemplate.hpp 给定了具体的实现接口，详情阅读其中的注释。Main.cpp 给出了一个测试，正确的 QM 算法必然会通过。也可以自己给出一些测试，帮助自己测试自己的算法的正确性。

下面对主要的接口以及期望的输出做一些说明。

```
static std::string QM(const unsigned int* m, unsigned int m_count,
                     const unsigned int* d, unsigned int d_count)
```

- 1、输出为输入的最小覆盖
- 2、输出为一个无空格的字符串如 $BC'D'+AC+AB'$ ，取反用英文单引号表示。
- 3、输出的字符串主要分成两个部分：本质本原蕴含项和其他项。每一部分都从布尔变量个数多到少排列，如示例中的本质本原蕴含项 $BC'D'+AC$
- 4、输出的每一项都是高位在前，低位在后，最高位为 A，然后依次 BCD……。
- 5、输出中除去本质本原蕴含项（也就是本原蕴含图中的剩余部分）的处理：先考虑输出项个数最少的，比如使用 ABC 或者 $AB+BC$ 就可以覆盖剩余最小项，那么输出 ABC ；其次考虑输出项个数相等的情况下，输出项中的布尔变量最少的，比如使用 $AB+BC$ 和 $BCD+AC$ 都可以，那么选择 $AB+BC$ 。

课件上的例子只使用 AB' 或者 AD' 就可以覆盖剩余项，所以不用 5 这条规则。

	4	8	10	11	12	15		=>	A	B	C	D
$m(4,12)$	X				X		-100	=>	-	1	0	0
$m(8,9,10,11)$		X	X	X			10--	=>	1	0	-	-
$m(8,10,12,14)$		X	X		X		1--0	=>	1	-	-	0
$m(10,11,14,15)$			X	X		X	1-1-	=>	1	-	1	-

$m(4,12)$ 和 $m(10,11,14,15)$ 是本质本原蕴含项

$$f(A,B,C,D)=BC'D'+AC+AB'/AD'$$