

Digital System Design Project 2

Exact Boolean Minimization

繳交期限：2022年10月13日23:59

以下翻译结果来自-有道神经网络翻译(NMT)

題目描述

积和(SOP)是2级布尔函数表示的标准形式。对应的字面量用乘积项进行and运算，然后一起进行or运算作为输出。SOP表示可以在AND-OR门或NAND-NAND门中实现。为了降低电路成本，重要的是尽量减少产品术语和字面量的数量。Maurice Karnaugh博士提出了一种真值表的图示形式，称为Karnaugh图，并递归地将相邻的正方形组合为素隐含(PI)代。然而，当布尔变量的数量大于6时，Karnaugh映射很难缩放。奎因博士和麦克卢斯基博士开发了一种基于计算机的表格形式，以有效地识别所有pi。这就是著名的奎因-麦克卢斯基算法。由于最小SOP表示是PI的子集，Petrick博士提出了一种从PI候选解中确定所有最小SOP解的技术。在这个项目中，您的C/ c++程序读取PLA格式的单一输出布尔函数(带don't cares)，并使用quni - mccluskey算法和Petrick算法以PLA格式输出其最小SOP表示。报告产品术语和字面量的数量。

请按以下规则提交报告和方案:

1. 报告的字体大小为12,PDF格式。
2. 报告的文件名是你的学生证(例如，B12345678.pdf)。
3. 用don't cares(4到6个变量)生成你自己的3个PLA文件。
4. 发布输入和输出PLA文件的内容。
5. 上传程序源代码的tarball(例如，B12345678.tgz)PLA文件和一个文本ReadMe文件，它说明了你的程序如何在Ubuntu Linux环境中编译和执行(例如，WSL)。

```
PLA Example: input.pla
.i 4
.o 1
.ilb a b c d
.ob f
.p7
00-0 1
0-11 1
1-01 1
0101 1
1111 -
100- 1
-01- 1
.e

SYNOPSIS
%> PROGRAM PLA_IN_FILE PLA_OUT_FILE
```

```
Run-time Example: %> mini input.pla output.pla
Total number of terms: 4
Total number of literals: 8

%> cat output.pla
.i4
.o1
.ilb a b c d
.ob f
.p4
-1-1 1
-0-0 1
1--1 1
--11 1
.e
```

參考:

[1] https://en.wikipedia.org/wiki/Quine%E2%80%93McCluskey_algorithm

[2] https://en.wikipedia.org/wiki/Petrick%27s_method

用don't cares(4到6个变量)生成你自己的3个PLA文件

```
input1.pla
1 .i 4
2 .o 1
3 .ilb a b c d
4 .ob f
5 .p 6
6 0-11 1
7 1-01 1
8 0101 1
9 1111 -
10 100- 1
11 -01- 1
12 .e
13
```

4變量

```
input2.pla
1 .i 5
2 .o 1
3 .ilb 甲 乙 丙 丁 戊
4 .ob f
5 .p 18
6 00000 -
7 00011 -
8 00101 -
9 00110 1
10 00111 -
11 01001 1
12 01010 -
13 01011 -
14 01110 1
15 01111 1
16 10100 -
17 10101 1
18 10111 1
19 11000 -
20 11001 -
21 11010 1
22 11101 1
23 11110 1
24 .e
25
```

5變量

```
input3.pla
1 .i 6
2 .o 1
3 .ilb 🙄 🙄 🙄 🙄 🙄 🙄
4 .ob f
5 .p 7
6 001000 1
7 0-1011 1
8 1-0101 1
9 11-100 1
10 111101 -
11 1-0111 1
12 -0111- 1
13 .e
14
```

6變量

输出PLA文件的内容

```
output1.pla
1 .i 4
2 .o 1
3 .ilb a b c d
4 .ob f
5 .p 3
6 -01- 1
7 10-- 1
8 -1-1 1
9 .e
10
```

```
output2.pla
1 .i 5
2 .o 1
3 .ilb 甲 乙 丙 丁 戊
4 .ob f
5 .p 5
6 -1001 1
7 1-101 1
8 -1-10 1
9 0-11- 1
10 -01-1 1
11 .e
12
```

```
output3.pla
1 .i 6
2 .o 1
3 .ilb 🙄 🙄 🙄 🙄 🙄 🙄
4 .ob f
5 .p 5
6 001000 1
7 0-1011 1
8 1-01-1 1
9 -0111- 1
10 11-10- 1
11 .e
12
```