

Logic System Assignment 1 Report

Department	Name	Student ID
EE112C	謝宗翰	E14084078

一、介紹

此次作業是目的是要將透過Karnaugh map來化簡邏輯函式，卡諾圖的目的是希望能最小化布林函數。但是，如果函式的變數 ≥ 5 時，有些項就很難圈了，Karnaugh map的行列數也將變得十分複雜。

故為了操作無限輸入變數的功能，此次程式的策略採用與Karnaugh map類似概念的Quine McCluskey method 與 Petrick's Method去實現化簡。

總共兩步驟：

1. 找出布林函數的所有 Prime implicants (Quine-McCluskey Algorithm)。
2. 使用 Prime implicants 來找出 minimum cover (Petrick's Method)。

二、程式流程

為了方便程式的撰寫，我的程式碼將分成3個部分撰寫：

1. Quine McCluskey method → 產生所有的Prime Implicant
2. Petrick's method → 提取 Min Cover
3. 主程式檔(Kmap.cpp)

1. 產生所有的Prime Implicant
1.1 Quine McCluskey重要函式講解
<ul style="list-style-type: none">● Dec_to_Bin(): 將十進位形式的minterm轉換成string 型態的二進位● sort_by_num_of_1(): 將二進位的minterm依照「1」的個數排列● sort_by_num_of_literal(): 將二進位的minterm依照「-」的個數排列● isGreyCode(): 檢查兩 Implicants 是否可以合併，可以則回傳true。例如：0110與0100即為 grey code。● Generate_Prime_Implic(): 如果 Implicant 經 isGreyCode 檢查不可合併，則視其為 Prime Implicant。● Solve(): 透過Generate_Prime_Implic()持續找出Prime Implicant，直到沒有Implicant可以合併為止。● Find_Prime_Implic_Pair(): 用bitwise "&"的方式，將二進位形式的 Prime Implicant 轉換成十進位minterm的形式，E.g., "0-0-"可轉換成(0,1,4,5)，以利接下來的Generate_essential()找出烙單項，進而得到Essential Prime Implicant● Generate_essential(): 列出所有Prime Implicant項的minterm，剔除don't care 項與重複項後，即可得Essential Prime Implicant。

A simple CAD tool based on K-map

2. 提取所有 Minimum Cover
<p>在簡化 Sum of Product 時有三大規則：</p> <ol style="list-style-type: none">1. $X + X = X$2. $XX = X$3. $X + XY = X$ <p>而程式碼內根據上述三大規則有不同的應對方法</p> <ul style="list-style-type: none">● $X + X = X$：在 <code>Expansion()</code> 中做檢查，如果發現已經出現過，便不會將其加入 SOP 中。● $XX = X$：利用容器 set 的特性，讓重複的元素只計算一次。使用 <code>vector< set<string> ></code> 來儲存 SOP，即可簡化這種情況。● $X + XY = X$：採取不處理的原則。此份程式碼的目的是尋找布林函數使用 Prime Implicant 來表達的 Minimum Cover，並非布林函數的最精簡表示式，因此沒有對這個規則做任何處理。

2.1 Petrick's Method重要函式講解
<ul style="list-style-type: none">● <code>Petrick_Method()</code>：使用 Quine McCluskey 產生的 Prime Implicants 以及 on-set 轉換成 Prime Implicant Chart 所對應的 Sum of Product(SOP)。● <code>Expansion()</code>：使用遞迴的方式展開將 POS 轉成 SOP 表達式。● <code>sort_by_SOP()</code>：讓 SOP 根據 clauses 數來排列● <code>Solve()</code>：將 SOP 函式展開後，取出 clauses 數量的 POS 即為答案。

3. 主程式檔(kmap.cpp) 操作
<ol style="list-style-type: none">1. 用 <code>loadinginput()</code> 函式將 "input.txt" 檔內的資料 load 進 Quine McCluskey 的 container 內，已運作 QM 演算法得 Prime Implicant 項和 Essential Implicant 項，最後將其輸出至 "output.txt" 內。2. 使用 Petrick class 中的 <code>Solve()</code> 可得最後的布林函數化簡，最後再將結果輸出到 "output.txt"。

p.s. 更詳細的程式碼解釋，可直接參照我的 .cpp 檔的註解

4. 總結流程
4.1 Quine McCluskey
<ol style="list-style-type: none">1. 透過 <code>sort_by_num_of_1()</code> 將所有 Implicant 依照「1」的個數分組來比較2. 使用 <code>Generate_Prime_Implic</code> 不停化簡至無可化簡的 Implicant 項<ol style="list-style-type: none">2.1 透過 <code>isGreyCode()</code> 比較每一組 Implicant 是否為 grey code<ol style="list-style-type: none">2.1.1 如果為 grey code 則進行 <code>Merge()</code> 融合成新的 Implicant 項
4.2 Petrick's Method
<ol style="list-style-type: none">1. 透過 <code>Expansion()</code> 函式展開 SOP 至 POS

A simple CAD tool based on K-map

2. 透過sort_by_SOP()提取出最簡項的POS
3. 將POS轉回SOP後輸出至"output.txt"

三、輸出結果

如何執行kmap.cpp：

將以下檔案放置同一資料夾後執行

- input.txt
- output.txt
- kmap.cpp
- Quine_McCluskey.cpp
- Quine_McCluskey.h
- Petrick_Method.cpp
- Petrick_Method.h

1. 1~4變數輸入	
1.1 三變數	
<div>input.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre> v 3 m 0,2,3,6 d 4,5,7 </pre>	<div>output.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre> \AB\ C \ 00 01 11 10 -----+-----+-----+-----+ 0 1 1 1 x -----+-----+-----+-----+ 1 0 1 x x -----+-----+-----+-----+ Prime Implicant = c',b,a Essential Prime Implicant = c',b The reduced SOP form: Answer 1 : F(A,B,C) = c'+b </pre>
1.2 四變數	
<div>input.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre> v 4 m 0,3,7,11,14 d 2,12,13,15 </pre>	<div>output.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre> \AB\ CD \ 00 01 11 10 -----+-----+-----+-----+ 00 1 0 x 0 -----+-----+-----+-----+ 01 0 0 x 0 -----+-----+-----+-----+ 11 1 1 x 1 -----+-----+-----+-----+ 10 x 0 1 0 -----+-----+-----+-----+ Prime Implicant = a'b'd',a'b'c,cd,ab Essential Prime Implicant = a'b'd',cd,ab The reduced SOP form: Answer 1 : F(A,B,C,D) = cd+a'b'd'+ab </pre>

2. 5~無限輸入 (5變數以上為簡潔無顯示Kmap)

2.1 五變數輸入

A simple CAD tool based on K-map

<div>input.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre>v 5 m 0,2,3,6,8,16,24,29,31 d 4,5,7,9,13,20</pre>	<div>output.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre>Prime Implicant = a'bc'd',bcd'e,abce,a'b'e',b'd'e',c'd'e',a'b'd,a'b Essential Prime Implicant = abce,c'd'e',a'b'd The reduced SOP form: Answer 1 : F(A,B,C,D,E) = c'd'e'+a'b'd+abce</pre>
<h3>2.2 十二變數輸入</h3>	
<div>input.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre>v 12 m 0,2,3,6,8,16,24,29,31,37,1058,1133,2999,4000,4021 d 4,5,7,9,13,20,1001,2222,3333</pre>	<div>output.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre>Prime Implicant = a'bc'd'e'f'gh'i'j'kl',a'bc'd'e'f'gh'ijk'l,abcdef'gh'i'j'k'l',ab'cdef'ghi'jkl,abcd 'jkl',a'b'c'd'e'f'g'h'ij'k',a'b'c'd'e'f'g'h'jkl',a'b'c'd'e'f'h'i'jkl',a'b'c'd'e' k'l',a'b'c'd'e'f'g'hijl,a'b'c'd'e'f'g'h'i'l',a'b'c'd'e'f'g'i'k'l',a'b'c'd'e'f'g'j a'b'c'd'e'f'g'h'i'k',a'b'c'd'e'f'g'h'i'j Essential Prime Implicant = a'bc'd'e'f'gh'i'j'kl',a'bc'd'e'f'gh'ijk'l,abcdef'gh'i'j'k'l',ab'cdef'ghi'jkl,abcd 'jkl',a'b'c'd'e'f'h'i'jkl',a'b'c'd'e'f'g'hijl,a'b'c'd'e'f'g'j'k'l',a'b'c'd'e'f'g The reduced SOP form: Answer 1 : F(A,B,C,D,E,F,G,H,I,J,K,L) = a'b'c'd'e'f'h'i'jkl'+a'b'c'd'e'f'g'j'k'l'+a'b'c'd'e'f'g'h'i'k'+a'b'c'd'e'f'g'hijl d'e'f'gh'i'j'kl'+a'b'c'd'e'f'gh'ijk'l+ab'cdef'ghi'jkl+abcdef'gh'i'j'k'l'+abcdef'gh</pre>

以上結果都與網路上的線上化簡器結果相符

<h3>3. 多種答案輸出</h3>																										
<div>input.txt - 記事本</div> <div>檔案 編輯 檢視</div> <pre>v 4 m 4,8,10,11,12,15 d 9,14</pre>	<div>output.txt - 記事本</div> <div>檔案 編輯 檢視</div> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th>AB\CD</th> <th>00</th> <th>01</th> <th>11</th> <th>10</th> </tr> </thead> <tbody> <tr> <th>00</th> <td>0</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <th>01</th> <td>0</td> <td>0</td> <td>0</td> <td>x</td> </tr> <tr> <th>11</th> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <th>10</th> <td>1</td> <td>0</td> <td>0</td> <td>x</td> </tr> </tbody> </table> <pre>Prime Implicant = bc'd',ab',ad',ac Essential Prime Implicant = bc'd',ac The reduced SOP form: Answer 1 : F(A,B,C,D) = bc'd'+ac+ab' Answer 2 : F(A,B,C,D) = bc'd'+ad'+ac</pre>	AB\CD	00	01	11	10	00	0	1	1	1	01	0	0	0	x	11	0	1	0	1	10	1	0	0	x
AB\CD	00	01	11	10																						
00	0	1	1	1																						
01	0	0	0	x																						
11	0	1	0	1																						
10	1	0	0	x																						
<p>如果有多個答案會顯示多種結果</p>																										