

Digital System Design Project 2 – Exact Boolean Minimization

姓名：張睿麟

學號：B11015030

系級：四資工二 乙

先備知識：

1. Prime Implicant、Essential Prime Implicant [1]
2. Quine–McCluskey algorithm [2]
3. Petrick's method [3]

實作步驟：

1. 把得到的 Terms 全部列出來
2. 利用 Quine–McCluskey algorithm 化簡 => 尋找 Essential Prime Implicant
 - 甲、化簡到最後沒辦法繼續化簡的 Terms 就是 Prime Implicant
 - 乙、尋找所有 Prime Implicant 中只有包含 1 個 minterm 的就是 Essential Prime Implicant
3. 利用 Petrick Algorithm 排列組合出所有的可能，進一步尋找最簡可能
 - 甲、針對非 Essential Prime Implicant 的 Prime Implicant 做排列組合
 - 乙、最短的組合再和一定需要的 EPI 組合成最簡方程式

程式實作說明：

儲存 Terms 的方式是用 string 資料型態

1. termsExpansion()

甲、利用遞迴把“-”分成 0、1，繼續遞迴直到讀完輸入的 terms(舉例：
10-1 => 1001、1011)

乙、遇到 Don't care 先看成 1，但是要記錄下來，因為在取 Essential Prime Implicant 時，不一定需要 Don't care

2. Classification()

甲、利用 for 迴圈及記錄每個 Terms 有多少個 1

乙、用 2D 的 vector 分類並儲存這些 Terms

3. Simplify()

甲、利用遞迴，把鄰近的 Terms 全部比對，若兩個 Terms 之間只差一個 bit，則把該 bit 替換成“-”進行化簡。

乙、在過程中，會去紀錄可以化簡的 Terms，化簡至該迴圈的最後，會去把沒辦法化簡的 Terms 新增至化簡好的表格，在進行化簡。

丙、所以程式結束後，能夠確保該表格已經不能化簡，且過程中沒辦法化簡的 Terms 也會儲存下來。

4. createPrimeImplicant()

甲、把化簡後的 Terms，展開，為了看哪些 minterms 有包含

乙、順便去紀錄所有的 minterms 出現過幾次

5. createEssentialPrimeImplicant()

甲、只有出現過 1 次的 minterms，可作為 Essential Prime Implicant

6. createPetrickTable()

甲、整理出非 Essential Prime Implicant 的表，稱做 othersTerms

7. 剩下的程式會把 otherTerms 的 terms 給上字母 a-z 標記為不同的 terms，利用 Petrick Algorithm 排列組合出所有的可能，因為是用 a-z 排列組合，所以可以針對每種組合的式子各自排序過後，遇到重複的字母，可以取一次就好。

8. 在針對每個式子做統整排序，遇到相同的式子可以刪除。

9. 剩下來的式子當中，長度最短的就可以成為最佳化簡的式子

測資演示：

4 Variables

Input1.pla	Output1.pla
<pre>.i 4 .o 1 .ilb a b c d .ob f .p 5 00-0 1 0-11 1 10-1 - 1-00 1 1111 1 .e</pre>	<pre>.i 4 .o 1 .lib a b c d .ob f .p 4 00-0 1 10-1 1 --11 1 1-00 1 .e</pre>
Terminal	
Total number of terms: 4	
Total number of literals: 11	

5 Variables

Input2.pla	Output2.pla
<pre>.i 5 .o 1 .ilb a b c d e .ob f .p 10 00-00 1 01-11 - -01-1 - 01001 1 0-110 1 01-1- 1 101-0 1 11-00 - 11001 - 110-0 1 .e</pre>	<pre>.i 5 .o 1 .lib a b c d e .ob f .p 6 -1001 1 -1010 1 01-1- 1 11-00 1 -01-- 1 00-00 1 .e</pre>
Terminal	
Total number of terms: 6	
Total number of literals: 21	

6 Variables

Input3.pla	Output3.pla
.i 6 .o 1 .ilb a b c d e g .ob f .p 10 10-1-0 1 0110-1 - 1-0101 1 11-1-0 1 -00-01 - 1-0111 1 -0111- - 1-011- 1 1001-- - -01110 - .e	.i 6 .o 1 .lib a b c d e g .ob f .p 5 -00-01 1 -0111- 1 0110-1 1 1--1-0 1 1-01-- 1 .e
Terminal	
Total number of terms: 5	
Total number of literals: 19	

參考資料：

[1]: <https://en.wikipedia.org/wiki/Implicant>

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

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