Digital System Design Project 1 – ROBDD Generation

姓名：張睿麟
學號：B11015030
系級：四資工二 乙

前言：

此次 Project 目標是要把不同輸入的真值表，依照給定的 Product term，以及化簡的方法，化簡出最終的答案，利用 Graphviz 畫出簡化過後的 Boolean function。

其中要撰寫一個程式能夠讀取 PLA，把用來儲存狀態表的 Tree 建立起來，依照 Product term 和兩個化簡的方法，輸出成 DOT 檔，而這個 DOT 檔是用來畫圖的。

化簡方法：

1. All isomorphic node => 化簡擁有相同結果的 then-edge、else-edge 的 node。
2. The redundeant node => 移除指向相同的 then-edge、else-edge 的 node。

ROBDD 程式邏輯：

1. 建立 struct node 的型別，用來儲存裡面 Tree 的節點。
2. 讀取並處理 PLA 檔案的資料。
3. 初始化我的 Tree。
4. 從 PLA 得到的真值表，寫出正確的 Tree，這部分需要用到 Recursion 遍歷需要更改值的 node。
5. 重複化簡，直到不能化簡為止。
6. 依照 DOT 需要的格式，寫檔案到 OUTPUT.dot 裡。

學習內容：

1. 要如何在 Linux 環境下編譯 Program 以及執行[1][2][3]：
   我是使用 C++編譯，那要先建置好環境，像是 g++的編譯器。
   建立好你的 Program 檔案，並且命名。
   在該路徑底下，執行「使用 g++編譯檔案」。
   $ g++ FILENAME
   接著輸入「檔案」、「要讀取的檔案」、「要寫入的檔案」
   $ ./a.out INPUT_FILENAME OUTPUT_FILENAME
   *這邊會有 INPUT_FILE 跟 OUTPUT_FILE 是因為 C 語言中，主程式會有參數輸入，就是指這個部分。
   ```
   int main(int argc, char *argv[])
   ```
2. 使用本機的 VS Code 遠端編譯 WSL 裡的程式[4]

Example 1 (4 variables):

PLA file
.i 4 .o 1 .ilb a b c d .ob F .p 5 -1-0 1 0-11 1 1100 1 0-00 1 1010 1 .e

Before Simplify:

Index	Variable	Then-edge	Else-edge	Comment
0	**0**			**Boolean 0**
1	**a**	**2**	**3**	
2	**B**	**4**	**5**	
3	**B**	**6**	**7**	
4	**C**	**8**	**9**	
5	**C**	**10**	**11**	
6	**C**	**12**	**13**	
7	**C**	**14**	**15**	
8	**D**	**0**	**16**	
9	**D**	**0**	**16**	
10	**D**	**0**	**16**	
11	**D**	**0**	**0**	
12	**D**	**16**	**16**	
13	**D**	**0**	**16**	
14	**D**	**16**	**0**	
15	**D**	**0**	**16**	
16	**1**			**Boolean 1**

After Simplify:

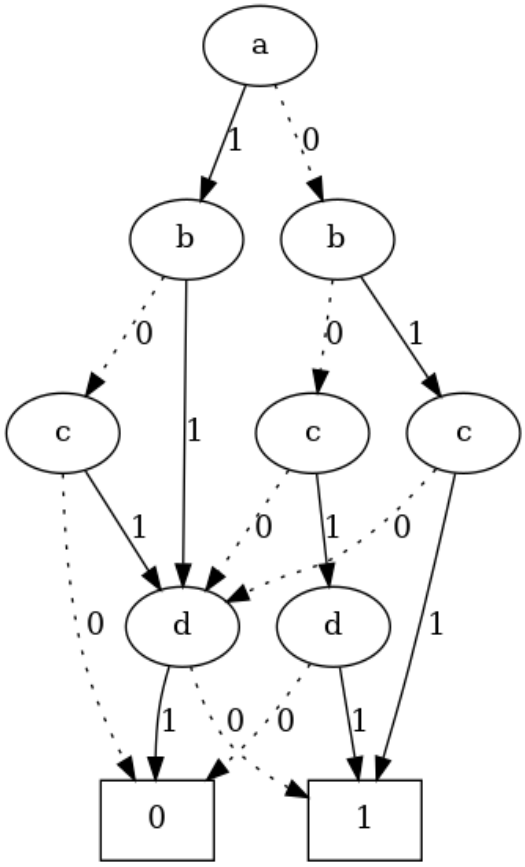
Index	Variable	Then-edge	Else-edge	Comment
0	**0**			**Boolean 0**
1	**a**	**2**	**3**	
2	**B**	**8**	**5**	
3	**B**	**6**	**7**	
4	C	8	8	Redundant
5	**C**	**8**	**0**	
6	**C**	**16**	**8**	
7	**C**	**14**	**8**	
8	**D**	**0**	**16**	
9	D	0	16	Redundant
10	D	0	16	Redundant
11	D	0	0	Redundant
12	D	16	16	Redundant
13	D	0	16	Redundant
14	**D**	**16**	**0**	
15	D	0	16	Redundant
16	**1**			**Boolean 1**

example1.dot

```
digraph ROBDD {
  {rank=same 1}
  {rank=same 2 3}
  {rank=same 5 6 7}
  {rank=same 8 14}

  0 [label="0", shape=box];
  1 [label="a"]
  2 [label="b"]
  3 [label="b"]
  5 [label="c"]
  6 [label="c"]
  7 [label="c"]
  8 [label="d"]
  14 [label="d"]
  16 [label="1", shape=box];

  1 -> 3 [label="0", style=dotted]
  1 -> 2 [label="1", style=solid]
  2 -> 5 [label="0", style=dotted]
  2 -> 8 [label="1", style=solid]
  3 -> 7 [label="0", style=dotted]
  3 -> 6 [label="1", style=solid]
  5 -> 0 [label="0", style=dotted]
  5 -> 8 [label="1", style=solid]
  6 -> 8 [label="0", style=dotted]
  6 -> 16 [label="1", style=solid]
  7 -> 8 [label="0", style=dotted]
  7 -> 14 [label="1", style=solid]
  8 -> 16 [label="0", style=dotted]
  8 -> 0 [label="1", style=solid]
  14 -> 0 [label="0", style=dotted]
  14 -> 16 [label="1", style=solid]
}
```



Example 2 (5 variables):

PLA file
.i 5 .o 1 .ilb a b c d e .ob F .p 5 0100- 1 -111- 1 1-001 1 01-10 1 011-1 1 .e

Before Simplify:

Index	Variable	Then-edge	Else-edge	Comment
0	**0**			**Boolean 0**
1	**a**	**2**	**3**	
2	**B**	**4**	**5**	
3	**B**	**6**	**7**	
4	**C**	**8**	**9**	
5	**C**	**10**	**11**	
6	**C**	**12**	**13**	
7	**C**	**14**	**15**	
8	**D**	**16**	**17**	
9	**D**	**18**	**19**	
10	**D**	**20**	**21**	
11	**D**	**22**	**23**	
12	**D**	**24**	**25**	
13	**D**	**26**	**27**	
14	**D**	**28**	**29**	
15	**D**	**30**	**31**	
16	**E**	**32**	**32**	
17	**E**	**0**	**0**	
18	**E**	**0**	**0**	
19	**E**	**32**	**0**	
20	**E**	**0**	**0**	
21	**E**	**0**	**0**	
22	**E**	**0**	**0**	
23	**E**	**32**	**0**	
24	**E**	**32**	**32**	
25	**E**	**32**	**0**	
26	**E**	**0**	**32**	
27	**E**	**32**	**32**	
28	**E**	**0**	**0**	
29	**E**	**0**	**0**	

30	**E**	**0**	**0**	
31	**E**	**0**	**0**	
32	**1**			**Boolean 1**

After Simplify:

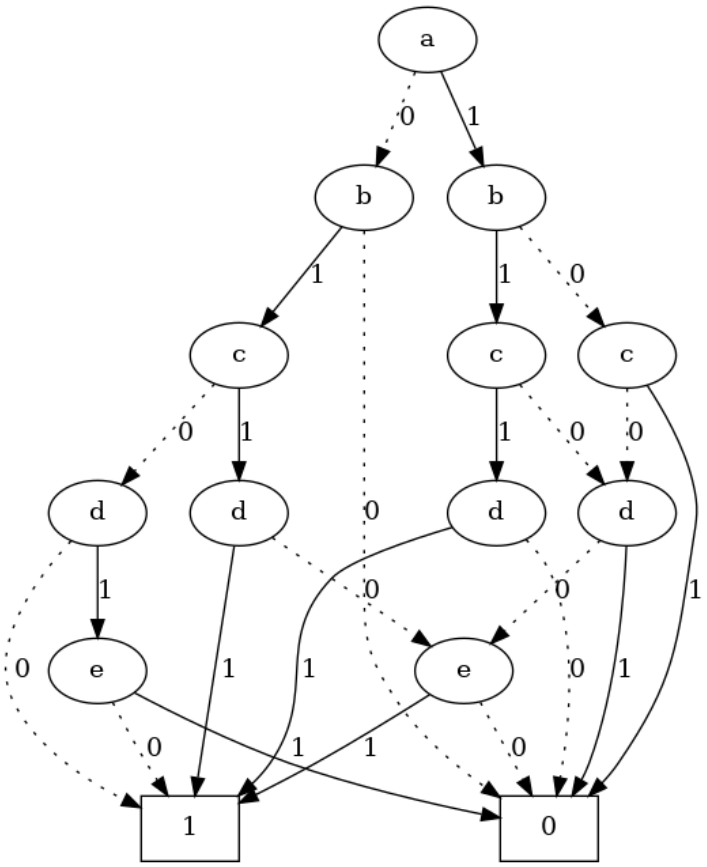
Index	Variable	Then-edge	Else-edge	Comment
0	**0**			**Boolean 0**
1	**a**	**2**	**3**	
2	**B**	**4**	**5**	
3	**B**	**6**	**0**	
4	**C**	**8**	**9**	
5	**C**	**0**	**9**	
6	**C**	**12**	**13**	
7	C	0	0	Redundant
8	**D**	**32**	**0**	
9	**D**	**0**	**19**	
10	D	0	0	Redundant
11	D	0	19	Redundant
12	**D**	**32**	**19**	
13	**D**	**26**	**32**	
14	D	0	0	Redundant
15	D	0	0	Redundant
16	E	32	32	Redundant
17	E	0	0	Redundant
18	E	0	0	Redundant
19	**E**	**32**	**0**	
20	E	0	0	Redundant
21	E	0	0	Redundant
22	E	0	0	Redundant
23	E	32	0	Redundant
24	E	32	32	Redundant
25	E	32	0	Redundant
26	**E**	**0**	**32**	
27	E	32	32	Redundant
28	E	0	0	Redundant
29	E	0	0	Redundant
30	E	0	0	Redundant
31	E	0	0	Redundant
32	**1**			**Boolean 1**

example2.dot

```
digraph ROBDD {
    {rank=same 1}
    {rank=same 2 3}
    {rank=same 4 5 6}
    {rank=same 8 9 12 13}
    {rank=same 19 26}

    0 [label="0", shape=box];
    1 [label="a"]
    2 [label="b"]
    3 [label="b"]
    4 [label="c"]
    5 [label="c"]
    6 [label="c"]
    8 [label="d"]
    9 [label="d"]
    12 [label="d"]
    13 [label="d"]
    19 [label="e"]
    26 [label="e"]
    32 [label="1", shape=box];

    1 -> 3 [label="0", style=dotted]
    1 -> 2 [label="1", style=solid]
    2 -> 5 [label="0", style=dotted]
    2 -> 4 [label="1", style=solid]
    3 -> 0 [label="0", style=dotted]
    3 -> 6 [label="1", style=solid]
    4 -> 9 [label="0", style=dotted]
    4 -> 8 [label="1", style=solid]
    5 -> 9 [label="0", style=dotted]
    5 -> 0 [label="1", style=solid]
    6 -> 13 [label="0", style=dotted]
    6 -> 12 [label="1", style=solid]
    8 -> 0 [label="0", style=dotted]
    8 -> 32 [label="1", style=solid]
    9 -> 19 [label="0", style=dotted]
    9 -> 0 [label="1", style=solid]
    12 -> 19 [label="0", style=dotted]
    12 -> 32 [label="1", style=solid]
    13 -> 32 [label="0", style=dotted]
    13 -> 26 [label="1", style=solid]
    19 -> 0 [label="0", style=dotted]
    19 -> 32 [label="1", style=solid]
    26 -> 32 [label="0", style=dotted]
    26 -> 0 [label="1", style=solid]
}
```



Reference:

[1]: https://stackoverflow.com/questions/46599733/how-to-read-file-from-argc

[2]: https://blog.gtwang.org/programming/c-cpp-tutorial-argc-argv-read-command-line-arguments/

[3]: https://www.cyberciti.biz/faq/howto-compile-and-run-c-cplusplus-code-in-linux/

[4]: https://code.visualstudio.com/docs/cpp/config-wsl