REPORT FOR COL215 ASSIGNMENT-2

Objective:

The objective of the assignment is to write a python program containing a function that takes arguments as two lists, namely func_TRUE and func_DC. It then returns the maximum possible expanded region for each term in func_TRUE.

Approach:

This objective can be achieved by first finding all the maximum possible regions. We observe that two regions can be combined to expand a region if they differ only by the boolean value of exactly one variable in the terms representing regions. This can be done by flipping a boolean value for each variable. Initially, we take a term from func_TRUE and get the number of variables and list of variables. Also, we have defined a helper function named lit_to_bool, which returns a list of Boolean values corresponding to each variable (E.g., 0 for a' and 1 for a). Now, we add all lists returned by the lit_to_bool function by passing elements of func_TRUE and func_DC to a set. We initialize a new empty set here.

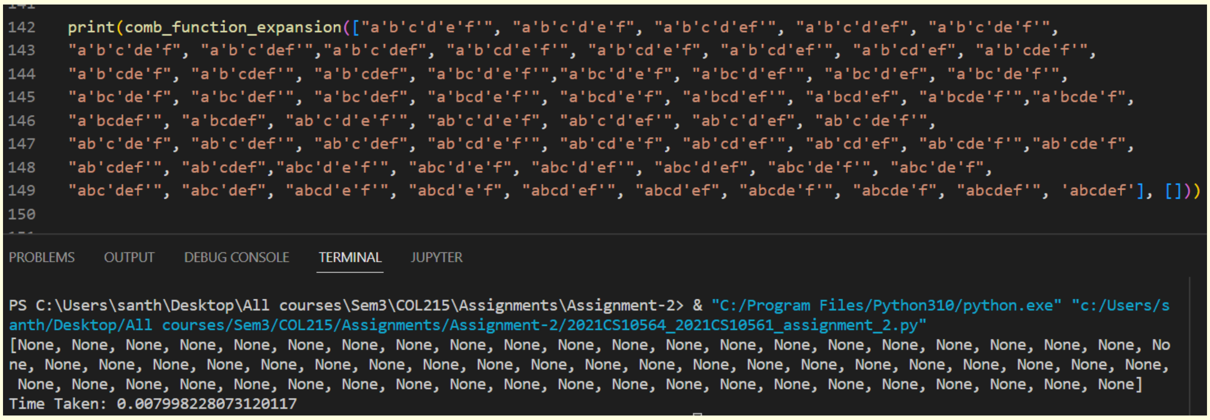
Now, we iterate over this set by taking each term and check if there exists a term that differs only by one position, which is done by flipping the boolean values of each variable. If there exists such a term, we add that term to a new set, with the value of the flipped variable as 'None'. If a term in the set doesn't contain any such term, it cannot be expanded anymore. So, we append it directly to the maxlist (which stores all the maximum possible regions). Now, by doing this, we have all possible distinct two cell regions in newset. We take this newset and go through the same iteration of checking in this newset. If these two cell regions have a term of possible two cell regions, then we can expand this term. This can be done by flipping each variable(except variables that have None) and checking if such a two-cell region exists. If such a two-cell region exists, then we expand the region by assigning the variable value to None and adding it to the new set. Similar to the previous iteration, we append these two cell regions if it doesn't have any region by which it can expand. We currently have all four cell regions in the list. Until there is no scope for expansion of any term, or all terms become None, these iterations occur.

We can see that the maximum number of times these iterations occur is n(number of variables) times, as every time either a term goes into the new list (which is indicated by change variable in code) or one of the values of variable changes to None. All variables become None by the end of n iterations(Worst case).

We obtained all maximum possible regions (including all 1's and x's) of the overall function. To get the maximum possible for each term of cells corresponding to 1's, we define a helper function terminregcheck, which checks if a particular cell is in the max region.

Testcases:

1. The Worst Case of 6-variables (All output terms are 1)



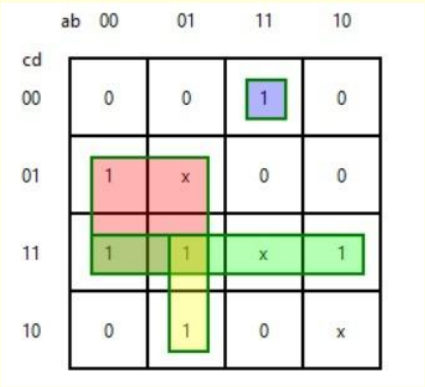
```
print(comb_function_expansion(["a'b'c'd'e'f'", "a'b'c'd'e'f", "a'b'c'd'ef'", "a'b'c'd'ef", "a'b'c'de'f'",
"a'b'c'de'f", "a'b'c'def'","a'b'c'def", "a'b'cd'e'f'", "a'b'cd'e'f", "a'b'cd'ef'", "a'b'cd'ef", "a'b'cde'f'",
"a'b'cde'f", "a'b'cdef'", "a'b'cdef", "a'bc'd'e'f'","a'bc'd'e'f", "a'bc'd'ef'", "a'bc'd'ef", "a'bc'de'f'",
"a'bc'de'f", "a'bc'def'", "a'bc'def", "a'bcd'e'f'", "a'bcd'e'f", "a'bcd'ef'", "a'bcd'ef", "a'bcde'f'","a'bcde'f",
"a'bcdef'", "a'bcdef", "ab'c'd'e'f'", "ab'c'd'e'f", "ab'c'd'ef'", "ab'c'd'ef", "ab'c'de'f'",
"ab'c'de'f", "ab'c'def'", "ab'c'def", "ab'cd'e'f'", "ab'cd'e'f", "ab'cd'ef'", "ab'cd'ef", "ab'cde'f'","ab'cde'f",
"ab'cdef'", "ab'cdef","abc'd'e'f'", "abc'd'e'f", "abc'd'ef'", "abc'd'ef", "abc'de'f'", "abc'de'f",
"abc'def'", "abc'def", "abcd'e'f'", "abcd'e'f", "abcd'ef'", "abcd'ef", "abcde'f'", "abcde'f", "abcdef'", 'abcdef'], []))
```

```
PS C:\Users\santh\Desktop\All courses\Sem3\COL215\Assignments\Assignment-2> & "C:/Program Files/Python310/python.exe" "c:/Users/s
anth/Desktop/All courses/Sem3/COL215/Assignments/Assignment-2/2021CS10564_2021CS10561_assignment_2.py"
[None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, No
ne, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None,
 None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]
Time Taken: 0.007998228073120117
```

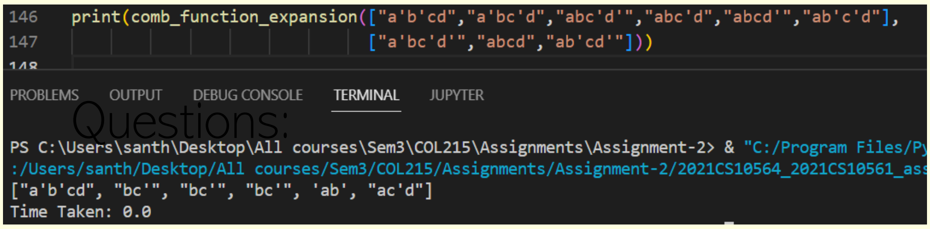
The output which must be the whole Kmap for each of the 2^8 (or 64) terms has been correctly computed by the program in a very short time. "1" represents that all variables can take any boolean value.

2. A case for 4-variables

As, we can see the kmap shown on the right can have many possible outputs. The program computes one of the possible outputs as shown below and highlighted.



cd \ ab	00	01	11	10
00	0	0	1	0
01	1	x	0	0
11	1	1	x	1
10	0	1	0	x



```
print(comb_function_expansion(["a'b'cd","a'bc'd","abc'd'","abc'd","abcd'","ab'c'd"],
                              ["a'bc'd'","abcd","ab'cd'"]))
```

Questions:

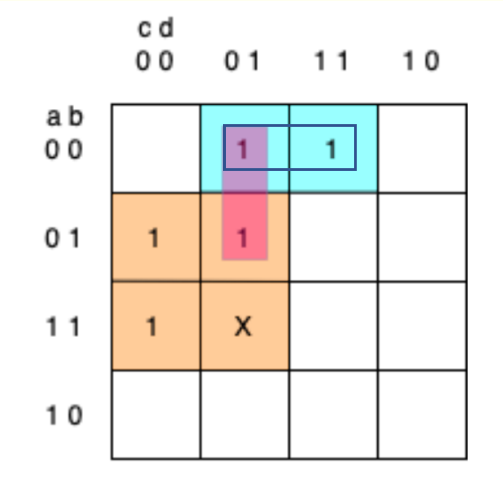
```
PS C:\Users\santh\Desktop\All courses\Sem3\COL215\Assignments\Assignment-2> & "C:/Program Files/Py
:/Users/santh/Desktop/All courses/Sem3/COL215/Assignments/Assignment-2/2021CS10564_2021CS10561_ass
["a'b'cd", "bc'", "bc'", "bc'", 'ab', "ac'd"]
Time Taken: 0.0
```

Questions:

Do all expansions result in an identical set of terms?

No, all expansions do not result in an identical set of terms. It is always possible that there are multiple regions of the same maximum possible size for a given cell say in FUNC_TRUE. In this case, there will be several sets of possible expanded regions. We can understand this through the below example.

In this case, the term a'b'c'd has two possible maximum regions namely a'b'd as well as a'c'd. Since both the regions have the same maximum size of 2 here, either of them can be taken to create the final set.

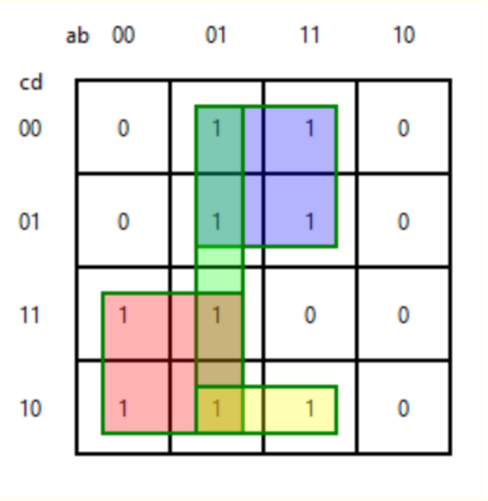


ab \ cd	00	01	11	10
00		1	1	
01	1	1		
11	1	X		
10				

Are all expansions equally good, assuming that our objective is to maximally expand each term? Explain.

Considering the update in the assignment, yes, all expansions are equally good since we do get a maximally expanded term corresponding to each input term. It does not matter which expanded region is taken as their maximum size is still the same. Each term will have grown as much as possible.

However, if our objective is to find maximally expanded regions and optimize the expression of a given function, then all expansions are not equally good. This is because there will be several overlaps and double counting of terms. Especially, when one of the expanded terms gets included in a few of the other terms. Let us understand this through the example shown.



cd \ ab	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	1	0	0
10	1	1	1	0

Here, the 4-cell vertical line a'b is redundant because it gets included in the other two squares. This makes the expansion inefficient.