COL215 - Software Assignment 2

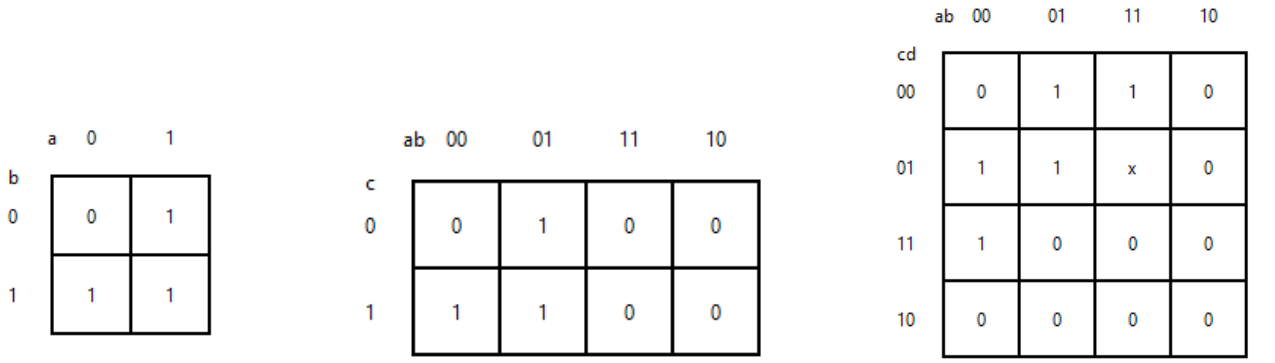
Boolean Function Expansion

Deadline: 6 September 2022

1 Introduction

In this assignment, we will follow up on the Karnaugh Map region formation by attempting to grow individual cells into regions enclosing only `TRUE(1)` and `DON'T CARE(x)`. Every expansion doubles the size of the region and leads to a literal being dropped from the corresponding term.

- Figure 1 (a): ab' can be expanded to also include ab (similarly, ab can be expanded to also include ab').
- Figure 1 (b): We have the opportunity to expand terms such as $a'bc'$, $a'b'c$, and $a'bc$.
- Figure 1 (c): Note that there are opportunities for expanding a cell multiple times. For example, $a'bc'd'$ can be expanded to obtain $bc'd'$, but after this, $bc'd'$ can be expanded further to obtain bc'.



(a) an example k-map of 2 variables

(b) an example k-map of 3 variables

(c) an example k-map of 4 variables

Figure 1: K-maps of different sizes

2 Problem Description

Given a function with 0s, 1s, and x's for each input combination, print the terms of the expanded function, where each cell is expanded to a legal region of maximum size.

- Expanded terms set should cover all the cells with '1'. No such condition for cells with 'x'.
- You can use the K-map to display the given function and expanded terms (if the number of variables is small enough), but your implementation should work for a function with any number of input variables.
- The algorithm cannot take time that is exponential in the number of input variables (however, it is OK if the results are not optimal for all inputs).

Answer the following questions in the report:

- Do all expansions result in an identical set of terms?
- Are all expansions equally good, assuming that our objective is to maximally expand each term? Explain.

2.1 Graphic Module

You are given a utility (`K_map_gui_tk.py`) that takes as input a function's output values for all input combinations, and displays the corresponding K-map. Get familiar with the utility. Sample usage of the utility is provided in the `test.py` file. You will need to work with functions of 2 to 4 variables for the assignment.

2.1.1 Instructions to use the Module

1. creating a K-map (L is a n * m 2D list):

```
root = kmap(L)
```

2. drawing region with $(x1, x2)$ as the top-left coordinate, $(x2, y2)$ as the bottom-right coordinate:

```
root.draw_regions(x1, y1, x2, y2, color)
# color can be 'red', 'green', 'blue', 'yellow' etc
```

3. display the K-map:

```
root.mainloop()
```

2.1.2 Using the Code for Wrapping Region

```
from K_map_gui_tk import *

root = kmap([[0,1,1,0], ['x',1,'x',0], [1,0,0,0], [1,'x',0,0]])
root.draw_region(1,3,2,0,'blue')
root.draw_region(3,0,0,3,'green')
root.mainloop()
```

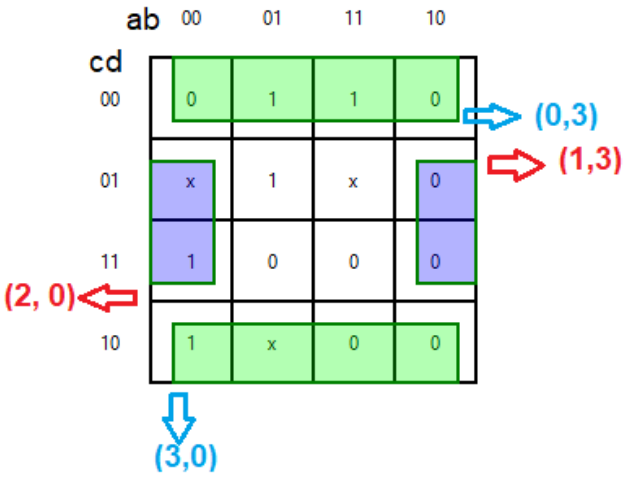


Figure 2: Output of the above code (arrows are for reference)

2.2 Test Cases

2.2.1 Sample Case I

```
Input:
func_TRUE = ["a'bc'd'", "abc'd'", "a'b'c'd", "a'bc'd", "a'b'cd"]
func_DC = ["abc'd"]

Output:
["bc'", "bc'", "a'c'd", "bc'","a'b'd"]
```

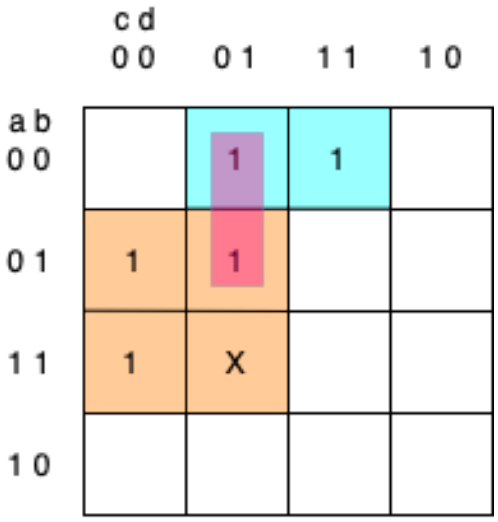


Figure 3: K-map for 4 variable

2.2.2 Sample Case II

```
Input:
func_TRUE = ["a'b'c'd'e'", "a'b'cd'e", "a'b'cde'", "a'bc'd'e'",
             "a'bc'd'e", "a'bc'de", "a'bc'de'", "ab'c'd'e'", "ab'cd'e'"]
func_DC = ["abc'd'e'", "abc'd'e", "abc'de", "abc'de'"]
Output:
["c'd'e'", "a'b'c'd'e'", "a'b'cde'", "bc'", "bc'", "bc'", "bc'", "c'd'e'", "ab'd'e'"]
```

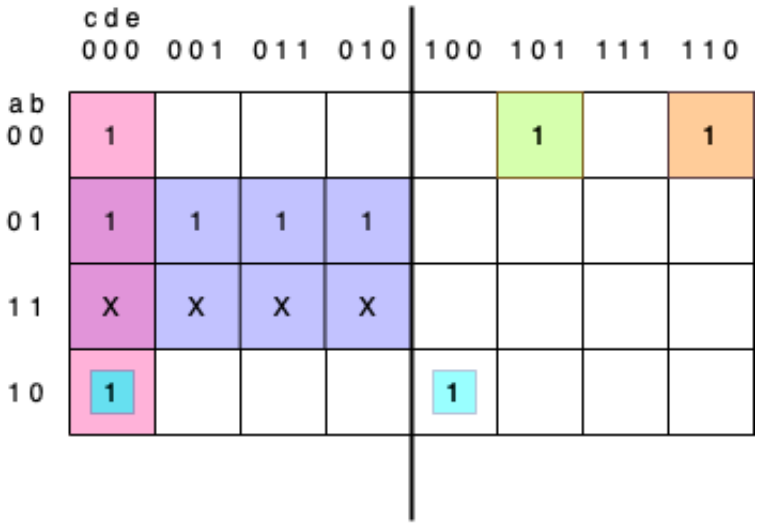


Figure 4: K-map for 5 variable

3 Submission Instructions

You are required to submit the following on Gradescope:

1. Python file named ⟨**entry_number1_number2**⟩**_assignment_2.py** containing the function

```
comb_function_expansion(func_TRUE, func_DC):
    """
        determines the maximum legal region for each term in the K-map function
        Arguments:
```

```
            func_TRUE: list containing the terms for which the output is '1'
            func_DC: list containing the terms for which the output is 'x'
        Return:
            a list of terms: expanded terms in form of boolean literals
    """
    # your code here
```

2. A short report (1-2 pages) explaining your approach and the test cases you have run.
3. Work in your group of 2 students for the assignment.
   - Groups will remain unchanged for subsequent assignments.
   - Only one submission per group is needed via gradescope. Gradescope help link: `https://help.gradescope.com/article/m5qz2xsnjy-student-add-group-members`
4. Post your doubts in the following thread on Piazza: `https://piazza.com/class/l6kqptxu7sz6i2/post/10`

3.1 Demo Instructions

In the Demo you will be asked to print intermediate results, i.e. for a given term print which all terms it combines to form the resulting term after expansion. Instead of legal terms you can also print the next legal region used in the expansion. Find below an example for the same-:

```
N = 3

Current term expansion: "a'b'"

Next Legal Terms for Expansion: "a'bc", "a'bc'"

Expanded Term: "a'"


N = 4

Current term expansion: "bd"

Next Legal Terms for Expansion: "a'bc'd'", "a'bcd'", "abc'd'", "abcd'"

Expanded Term: "b"


N = 6

Current term expansion: "d'ef"

Next Legal Terms for Expansion: "a'b'c'def" , "a'bcdef" , "ab'cdef" , "abc'def",
"a'b'cdef" , "a'bc'def" , "ab'c'def" , "abcdef"

Expanded Term: "ef"


N = 8

Current term expansion: "bc'e'gh"

Next Legal Terms for Expansion: "a'bc'defgh" , "a'bc'd'efgh" , "abc'd'efgh" , "abc'defgh" , "a'bc'def'gh" ,
"a'bc'd'ef'gh" , "abc'd'ef'gh" , "abc'def'gh"

Expanded Term: "bc'g"
```