

# COL215 - Software Assignment 1

## Karnaugh Maps

Submission Deadline: 21 Aug 2022

### 1 Introduction

In this assignment we will construct and manipulate Karnaugh Maps (K-maps). Examples of K-Maps are shown below (Figure 1), representing a function of 2, 3, 4 boolean/binary variables  $a$ ,  $b$ ,  $c$ , and  $d$ . For each  $abcd$  combination, the function can have one of the following values: TRUE (1), FALSE (0), and DON'T CARE ( $x$ ). DON'T CARE implies that the output value of the function is not relevant, so can be either 0 or 1.

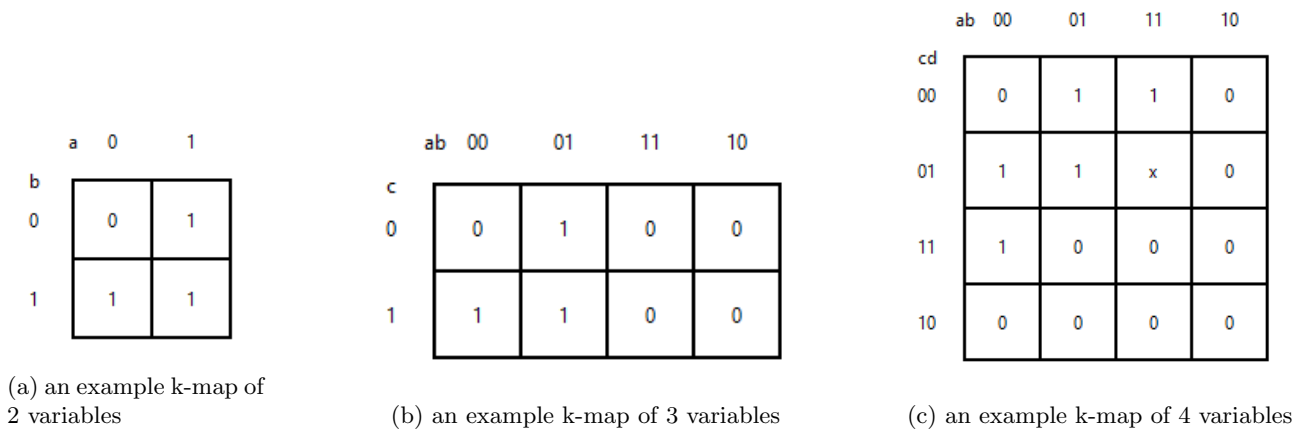


Figure 1: K-maps of different sizes

### 2 Problem Description

For a K-map with 4 variables, the rows of the K-map are labelled with  $ab$  values, and columns are labelled with  $cd$  values. Note the sequence of values [00, 01, 11, 10]. This is a K-map property (adjacent cells have only one variable changing values).

A REGION in the K-map is identified by two co-ordinates: the top-left corner, and bottom-right corner (note that the regions can wrap around the edges). Regions correspond to TERMS (e.g.,  $ab'c$ ), which are the product of LITERALS. Literals can be a boolean variable or its complement (e.g.,  $a$  or  $b'$ ). Here is the way to find the region corresponding to a term.

- If a variable  $a$  appears in *uncomplemented form* (e.g.,  $ab$ ), then the region has  $a = 1$  in all its cells.
- If a variable  $a$  appears in *complemented form* (e.g.,  $a'b$ ), then the region has  $a = 0$  in all its cells.
- If a variable  $k$  does not appear in the term, then the region has both cells with  $k = 0$  and cells with  $k = 1$ .

Given a function and a term, write a program to:

- highlight the corresponding K-map region, and
- report whether the region is LEGAL. A legal region can consist of 1s and x's, but cannot contain any 0s.

#### 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 \* n 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()
```

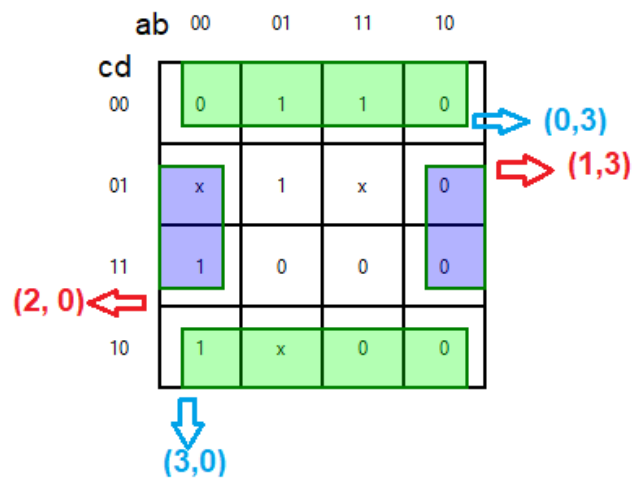


Figure 2: Output of the above code (arrows are for reference)

## 2.2 Example Test Case

Refer to Figure 3 for an example on the correspondence between terms/regions and how to identify LEGAL and ILLEGAL regions.

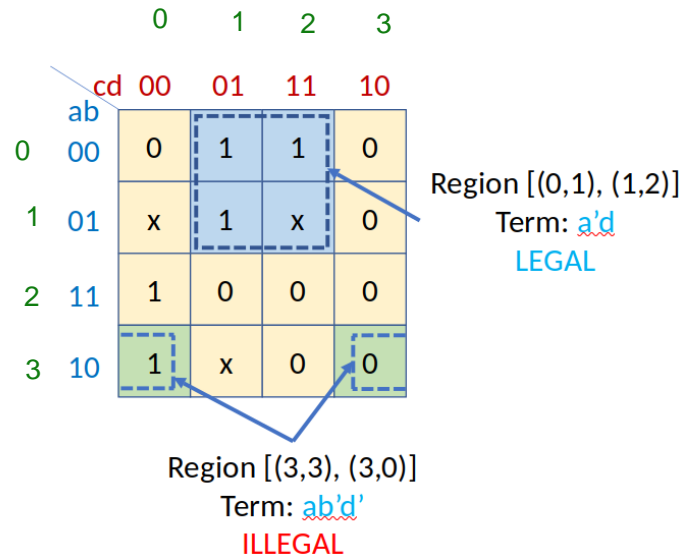


Figure 3: K-map with two highlighted regions: one is legal, the other is not

The region specified by [(0,1), (1,2)] (term:  $a'd$ ) is **LEGAL** since it only has 1 and  $x$ . However, the region [(3,3), (3,0)] (term  $ab'd'$ ) is **ILLEGAL** since it also has a 0.

### 3 Submission Instructions

You are required to submit the following on Gradescope:

1. Python file named `<entry_number1_number2>_assignment_1.py` containing the function

```
is_legal_region(kmap_function, term):
    """
    determines whether the specified region is LEGAL for the K-map function
    Arguments:
        kmap_function: n * m list containing the kmap function
        for 2-input kmap this will 2*2
        3-input kmap this will 2*4
        4-input kmap this will 4*4
        term: a list of size n (term[i] = 0 or 1 or None,
        corresponding to the i-th variable)
    Return:
        three-tuple: (top-left coordinate, bottom right coordinate, boolean value)
        each coordinate is represented as a 2-tuple
    """
    # your code here
```

2. A short report (1-2 pages) explaining your approach and the test cases you have run.
3. You have to form a group of 2 students for the assignment.
  - Group will remain same for subsequent assignment, no change allowed.
  - Only one submission per group needed to be done via gradescope. Gradescope help link: <https://help.gradescope.com/article/m5qz2xsnjy-student-add-group-members>
4. Post your doubt in the following thread on Piazza: <https://piazza.com/class/l6kqptxu7sz6i2?cid=8>