COL215P: Software Assignment-1: Karnauph Maps

Ishita Chaudhary 2019CS10360
Kshitiz Bansal 2019CS50438

Problem Description

Given a K map and a term, write a program to report whether the region corresponding to this term is LEGAL and return the top-left and bottom-right coordinates of the region as well. A legal region can consist of 1s and x's, but cannot contain any 0s.

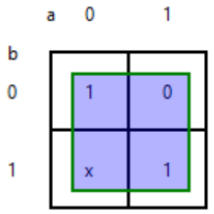
Approach

We create row and column headers in gray code format based on the number of variables (in the input term). As discussed on Piazza, the first n - n//2 variables correspond to columns, and the following n//2 variables correspond to the rows.
After this, we convert the input term (which is a list of 0/1/None) to a string (like, 10x0; x corresponds to "do not care"). We do this because it becomes easier to compare with row/column headers in the next step.
Then we find the extremes of the region for the given term. We iterate twice (in both directions) to find the extremes and then check if the region between them is the required region or the region outside them. This is done for both, the vertical and horizontal directions.
After we have found the extremes ((x1, y1), (x2, y2)), we have to check if the region is legal; we iterate over the matrix to check if any zeros occur in the region. If they do, then it is illegal, otherwise legal.

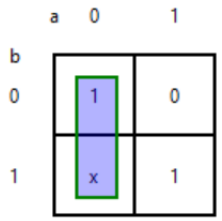
Testing

We tested the above-described implementation for K-maps with 2, 3, and 4 variables. We covered the cases which verify the wrapping property of K-maps. We also tested the code on all possible sizes of the regions. In each test case given below, the term describes the region, shaded in blue in the corresponding K-map, and the output of the code. The NULL term means each variable could take any value (0 or 1) and covers the complete K-map.

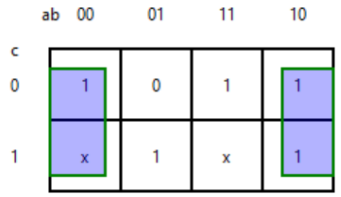
- For two variables
    - Testcase-1: term= NULL, output = ((0, 0), (1, 1), False)



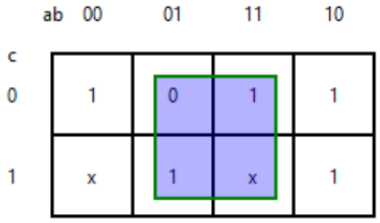
- Testcase-2: term= a', output = ((0, 0), (1, 0), True)



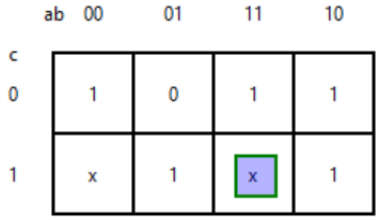
- For three variables
  - Testcase-1: term= b', output = ((0, 3), (1, 0), True)



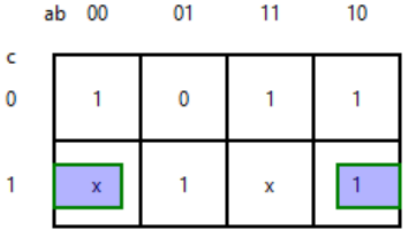
  - Testcase-2: term= b, output = ((0, 1), (1, 2), False)



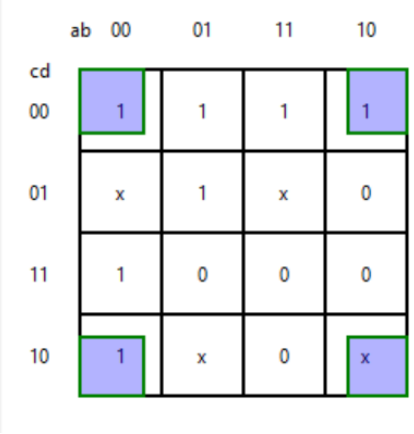
  - Testcase-3: term= abc, output = ((1, 2), (1, 2), True)



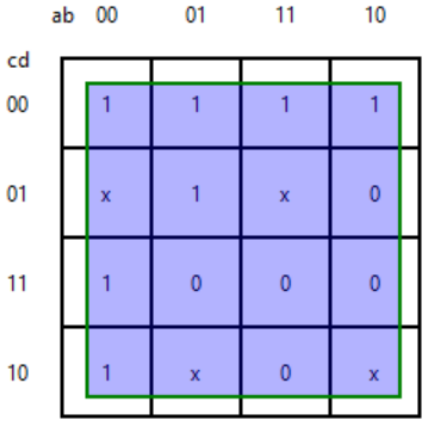
  - Testcase-4: term= b'c, output = ((1, 3), (1, 0), True)



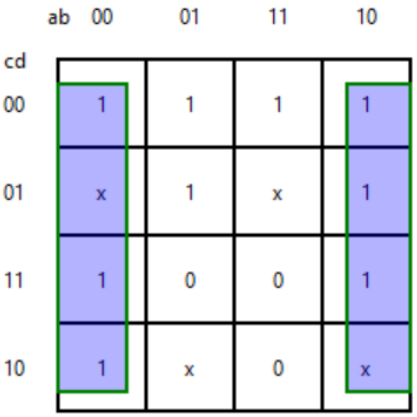
- For four variables
    - Testcase-1: term= b'd', output = ((3, 3), (0, 0), True)



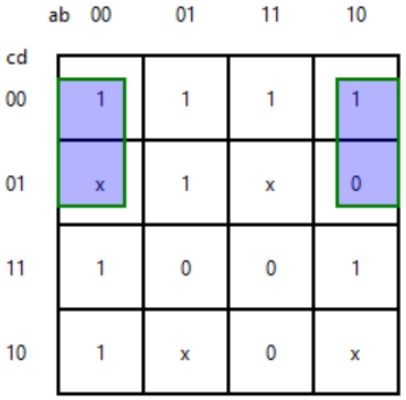
    - Testcase-2: term= NULL, output = ((0, 0), (3, 3), False)



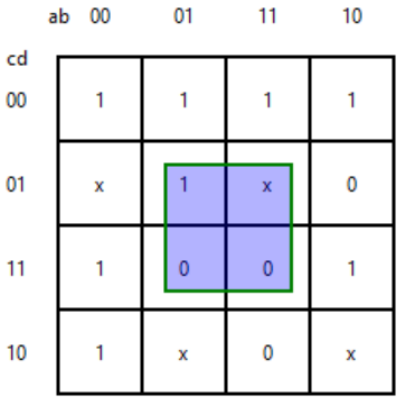
    - Testcase-3: term= b', output = ((0, 3), (3, 0), True)



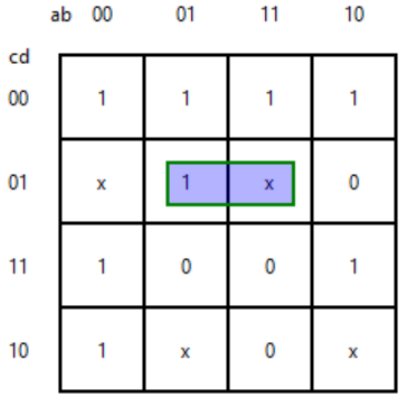
- Testcase-4: term= b'c', output = ((0, 3), (1, 0), False)



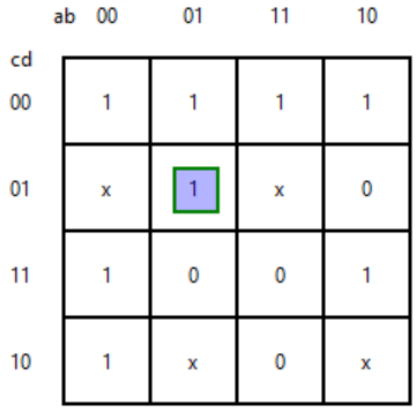
- Testcase-5: term= bd, output = ((1, 1), (2, 2), False)



- Testcase-6: term= bc'd, output = ((1, 1), (1, 2), True)



- Testcase-7: term=a'bc'd, output = ((1, 1), (1, 1), True)



- Testcase-8: term= bd', output = ((3, 1), (0, 2), True)