

COL215 Assignment 1

- Report by Piya Bundela(2021CS10118) and Aastha Rajani(2021CS10093)

Assignment 1 of col215 makes us use knowledge of K-maps. Here we are provided list of lists of values which forms a k-map and are supposed to give coordinates of given region in corresponding k-map and judge if given region is legal. A Karnaugh map (K-map) is a visual method used to simplify the algebraic expressions in Boolean functions without having to resort to complex theorems or equation manipulations. A legal region can be described as region that can consist of 1s and x's, but cannot contain any 0s.

Assignment has been done in three cases (case1 - 2 variables , case2 - 3 variables , case3 - 4 variables)

To begin with, we started with converting given list of lists into one list named Kmap_function . We have then defined 3 dictionaries that maps coordinate of k-maps with corresponding indices in the list Kmap_function for each case.

Depending on the number of unknown variables, term will be given (length of term = number of unknowns). Each element of term will be describing a condition for region. So, we have created lists which will contain coordinates of region as described by each condition. Final region will contain coordinates which are common after applying all condition.

These coordinates are then sorted.

In case of two variables, we have two cases.

Case1- When the common region contains only one single coordinate. In that case we have returned the same coordinate twice.

Case2- In all other cases, we return the topmost left and bottommost right coordinates.

In case of three variables, we have three cases.

Case1- Wrapping occurs only when the common points belong only to the zeroth and third column of the kmap. So, in order to identify this, we have checked whether the list of common coordinates satisfy this condition, and if it does we have given the coordinates at the first indexes from start and last in our sorted list of common coordinates.

Case1- Wrapping occurs only when the common points belong only to the zeroth and third column of the kmap. So, in order to identify this, we have checked whether the list of common coordinates satisfy this condition, and if it does we have given the first coordinate as the topmost right and then the bottom most left of the common region.

Case2- When the common region contains only one single coordinate. In that case we have returned the same coordinate twice.

Case3-In all other cases, we need to display the bottom most left and top most right coordinates.

Similarly, for the case with four variables, we have four different cases.

Case1- when the common region contains only one single coordinate. In that case we have returned the same coordinate twice.

Case2-This is a special case when the common coordinates are the corner coordinates of the kmap .In that case we need to return (0,0) and (3,3) as output.

Case3-This is a case of column wrapping, i.e., when the common coordinates belong only to zeroth and third column. We have identified this case and returned first coordinate as the topmost right and then the bottom most left of the common region.

Case 4-This is the case of row wrapping, i.e., when the common coordinates belong only to zeroth and third row. We have identified this case and returned first coordinate as the bottommost left and then the topmost right of the common region.

Simultaneously, we have also extracted the values in k maps from Kmap_function on these coordinates using the dictionary we created to map coordinates with indices. To check if the region we have created is legal, we will check if these extracted values contain 0. (Region is legal if there is no 0).

The function will report answer in the form [corresponding coordinates 1, corresponding coordinates 2, True if region is legal/False otherwise).

Test Cases

```
#For 2 variable k map
k1=[[1, 'x'], [0,1]]
t1=is_legal_region(k1, [None, None])
t2=is_legal_region(k1, [1,1])
t3=is_legal_region(k1, [0, None])

# For 3 variable kmap
k2=[[1, 'x', 0, 1], ['x', 1, 0, 'x']]
t4=is_legal_region(k2, [None, 0, None])
t5=is_legal_region(k2, [None, None, None])
t6=is_legal_region(k2, [0, 1, 0])

# For 4 variable kmap
k3 = [[1, 0, 'x', 0], [1, 1, 1, 0], [1, 'x', 'x', 0], [1, 0, 1, 0]]
t7=is_legal_region(k3, [None, 0, None, 0])
t8=is_legal_region(k3, [None, None, None, 0])
t9=is_legal_region(k3, [1, 1, 0, 0])
t10=is_legal_region(k3, [None, None, None, None])
```

((0, 0), (1, 1), False)

((1, 1), (1, 1), True)

((0, 0), (1, 0), False)

((0, 3), (1, 0), True)

((0, 0), (1, 3), False)

((0, 1), (0, 1), True)

((3, 3), (0, 0), False)

((3, 0), (0, 3), False)

((0, 2), (0, 2), True)

((0, 0), (3, 3), False)

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Kmaps_code.py - Kmaps_code.py
assignment_1 Kmaps_code.py
Kmaps_code.py
285
286 #For 2 variable k map
287 k1=[['x'],[0,1]]
288 t1=is_legal_region(k1,[None,None])
289 t2=is_legal_region(k1,[1,1])
290 t3=is_legal_region(k1,[0,None])
291
292 # For 3 variable kmap
293 k2=[['x','x',0,1],['x','x',1,0,'x']]
294 t4=is_legal_region(k2,[None,None])
295 t5=is_legal_region(k2,[None,None,None])
296 t6=is_legal_region(k2,[0,1,0])
297
298 # For 4 variable kmap
299 k3 = [['x','x','x',0], [1,1,1,0],['x','x','x',0],['x',0,1,0]]
300 t7=is_legal_region(k3,[None,0,None,0])
301 t8=is_legal_region(k3,[None,None,None,0])
302 t9=is_legal_region(k3,[1,1,0,0])
303 t10=is_legal_region(k3,[None,None,None,None])
304
305 print(t1)
306 print(t2)
307 print(t3)
308 print(t4)
309 print(t5)
310 print(t6)
311 print(t7)
312 print(t8)
313 print(t9)
314 print(t10)
PEP 8: W292 no newline at end of file
314:11 CRLF UTF-8 4 spaces Python 3.10 (assignment_1)
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Kmaps_code.py - Kmaps_code.py
assignment_1 Kmaps_code.py
Kmaps_code.py
302 t9=is_legal_region(k3,[1,1,0,0])
303 t10=is_legal_region(k3,[None,None,None,None])
304
305 print(t1)
306 print(t2)
307 print(t3)
308 print(t4)
309 print(t5)
310 print(t6)
311 print(t7)
312 print(t8)
313 print(t9)
314 print(t10)
Run: Kmaps_code C:\Users\admin\PycharmProjects\assignment_1\env\Scripts\python.exe C:/Users/admin/PycharmProjects/assignment_1/Kmaps_code.py
((0, 0), (1, 1), False)
((1, 1), (1, 1), True)
((0, 0), (1, 0), False)
((0, 3), (1, 0), True)
((0, 0), (1, 3), False)
((0, 1), (0, 1), True)
((3, 3), (0, 0), False)
((3, 0), (0, 3), False)
((0, 2), (0, 2), True)
((0, 0), (3, 3), False)
Process finished with exit code 0
Download new built shared indexes: Post-are the indexes time and CBI load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... 161 CRLF UTF-8 4 spaces Python 3.10 (assignment_1)
```