COL 215 SW A1

Rajat Bhardwaj
2020CS50436
Getaansh Juneja
2020CS50649

October 2022

1 Report

1.1 GreyCode

We have implemented a function that can generate the greycode for n - bits. Since we require to generalise(extend) the legality checking to n-variables, we have implemented this function. The size of the list is 2^n. The algorithm of the function is as follow.

We will do this recursively. Let the solution for n be L. Now we reverse L and append it with itself. Thus we get a list with list of size 2n with nth and (n+1) elements equal and this list is a palindrome as we can see clearly. Now we just append 0 to first n element and 1 to last n elements thus we get a grey code of (n+1) terms. The base case will be for 1 term i.e. [0,1].

Since we can generate the greycode for any number of terms like this, we can check the legality of any number of terms.

1.2 Checking legality with Greycode

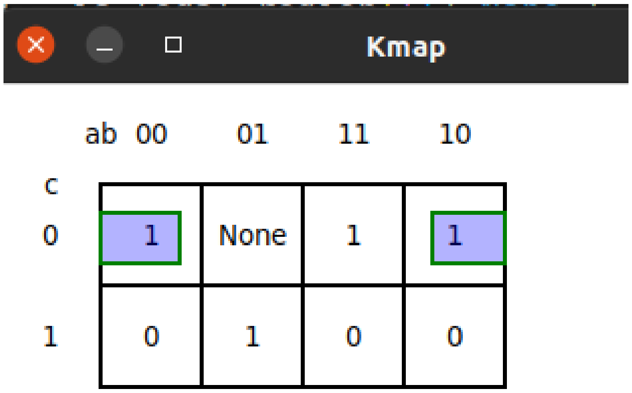
We created a dictionary that mapped the Greycode with the value of the terms. Then we iterate in the dictionary, if the coordinated satisfy the condition of the input then we check the value in the k-map, if the value is 0 for any element we will make the flag False, else the flag will remain true. We return this flag in the end. The function can return the legality for any general value of n.

1.3 Region Output

We maintain a list **region** which contains all the coordinates of the cell corresponding to the given input region. Then we output top-left and bottom right coordinate according to the size of the kmap i.e number of rows **n** and number of columns **m**. The following are the cases:

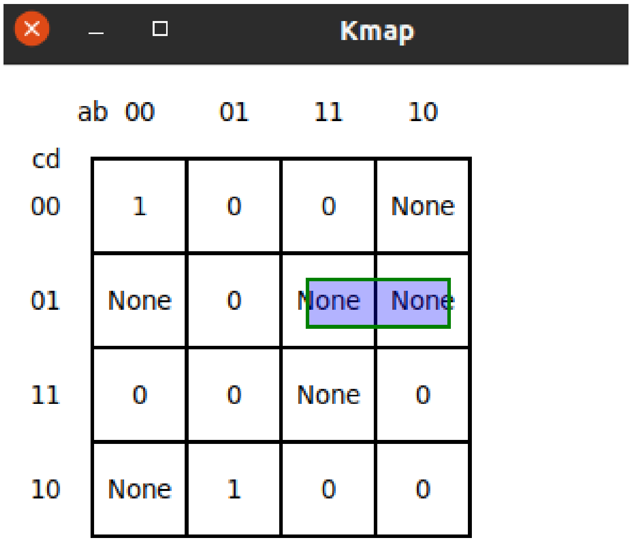
1. n=4 and m=4: We further have cases based on the length of the list region. In that we further check whether the coordinates coincides with the edges of the kmap or not and finally output the top-left and bottom-right coordinate accordingly.
2. n=2 and m=4: This case is similar to the above case only. We first check length of the region and then further check whether the coordinates coincides with the edges of the kmap or not and finally output the top-left and bottom-right coordinate accordingly.
3. n=2 and m=2: In this case we simply output first coordinate, last coordinate in list region and a boolean value indicating whether the region is legal or not.

1.4 Test Cases



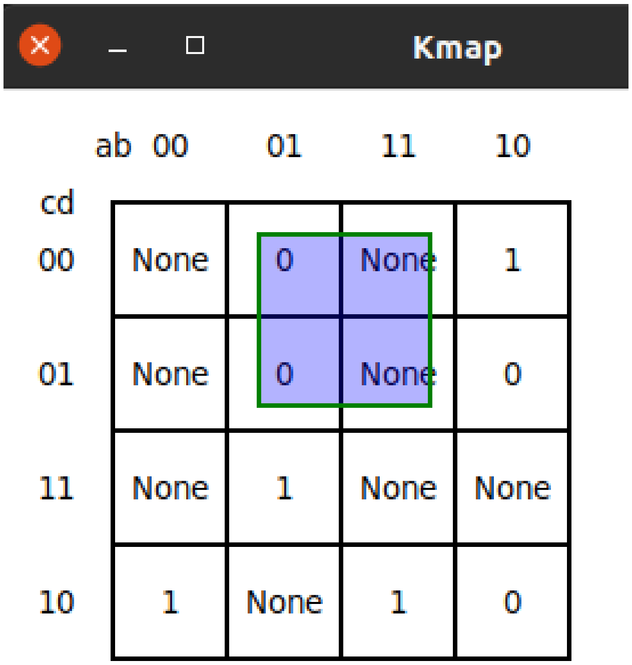
ab	00	01	11	10
c				
0	1	None	1	1
1	0	1	0	0

Figure 1: is_legal_region([[1,None,1,1],[0,1,0,0]], [None,0,0])



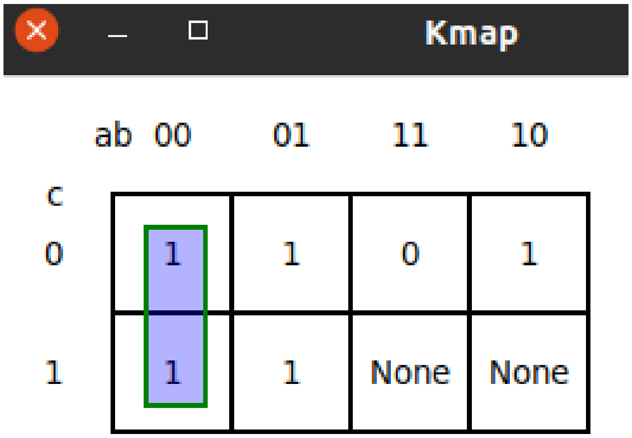
ab	00	01	11	10
cd				
00	1	0	0	None
01	None	0	None	None
11	0	0	None	0
10	None	1	0	0

Figure 2: is_legal_region([[1,0,0,None],[None,0,None,None],[0,0,None,0],[None,1,0,0]],[1,None,0,1])



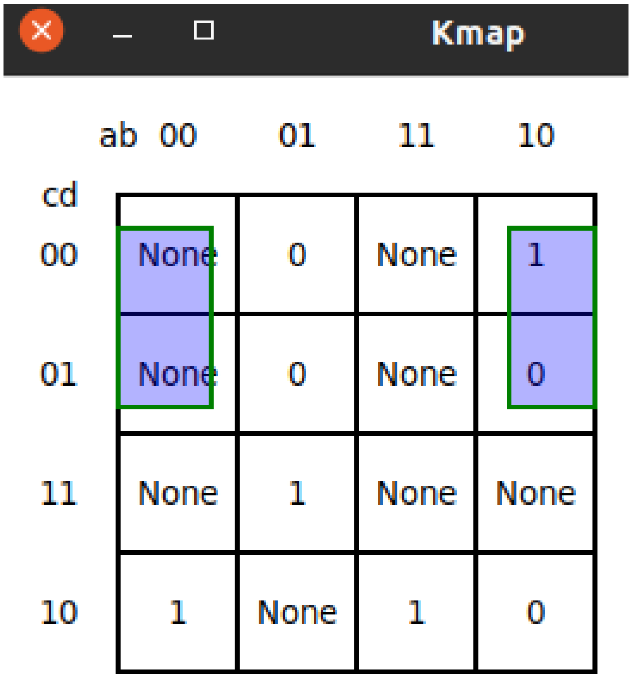
ab cd	00	01	11	10
00	None	0	None	1
01	None	0	None	0
11	None	1	None	None
10	1	None	1	0

Figure 3: is_legal_region([[None,0,None,1],[None,0,None,0],[None,1,None,None],[1,None,1,0]],[None,1,0,None])



ab c	00	01	11	10
0	1	1	0	1
1	1	1	None	None

Figure 4: is_legal_region([[1,1,0,1],[1,1,None,None]],[0,0,None])



ab cd	00	01	11	10
00	None	0	None	1
01	None	0	None	0
11	None	1	None	None
10	1	None	1	0

Figure 5: is_legal_region([[None,0,None,1],[None,0,None,0],[None,1,None,None],[1,None,1,0]],[None,0,0,None])