Srijan Gupta - 2020CS50444, Nischay Diwan - 2020CS50433
COL215P: Digital Logic & System Design

Legal Region in a Karnaugh map

Approach

As specified in the assignment instructions, our function only supports finding regions corresponding to minterms when the number of variables is 2,3 or 4, and, once the region is found, the logic to check its legality can be scaled to any number of inputs.
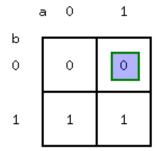
We check for each literal value in term given, case by case to narrow down highlighted kmap region. For eg in case of kmap of 4 x 4, with term be bc'd, here value of a is none, b is 1, c is 0, and d is 1. Now we narrow down both x and y coordinates separately, 3 * 3 = 9 cases for ab, and 3 * 3 = 9 cases for cd. Some cases maybe redundant but we still placed them for uniformity and understanding.

After this, once we get both the coordinates, the code for checking legality is generic for any m,n. It is simple "nested for loop" broken in cases depending on the wrap around condition.
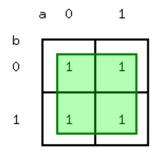
Tests

The following test cases are exhauative for types of highlighted region, and same are commented in code. Green ones are legal and blue are illegal

1. Tests for 2 x 2

1.1. [[0,0],[1,1]],[1,0], Term is ab'

1.2. [[1,1],[1,1]],[None,None], Term is None

1.3. [[0,1],[1,1]],[1,None], Term is a

1.4. [[0,1],[1,1]],[None,0], Term is b'

2. Tests for 2 x 4

2.1. [[0,1,1,0],[1,1,'x',0]],[None,1,None], Term is b

2.2. [[0,1,1,0],[1,1,'x',0]],[None,None,None],Term is None

2.3. [[0,1,1,0],[1,1,'x',0]],[0,1,1],Term is a'bc

2.4. [[0,1,1,0],[1,1,'x',0]],[None,0,0],Term is b'c'

c \ ab	00	01	11	10
0	0	1	1	0
1	1	1	×	0

2.5. [[0,1,1,0],[1,1,'x',0]],[None,0,None], Term is b'

c \ ab	00	01	11	10
0	0	1	1	0
1	1	1	×	0

2.6. [[0,1,1,0],[1,1,'x',0]],[1,1,None],Term is ab

c \ ab	00	01	11	10
0	0	1	1	0
1	1	1	×	0

3. Tests for 4 x 4
3.1 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[1,0,0,1], Term is ab'c'd

cd \ ab	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.2 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[0,1,None,None], Term is a'b

cd \ ab	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.3 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,None,1,0], term is cd'

ab \ cd	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.4 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,1,None,1])), term is bd

ab \ cd	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.5 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,1,None,0])), term is bd'

ab \ cd	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.6 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,0,None,0])), term is b'd'

ab \ cd	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.7 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,0,None,1])), term is b'd

cd \ ab	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0

3.8 [[0,1,1,0],[1,1,'x',0],[1,0,0,0],[0,0,0,0]],[None,1,0,None])), term is bc'

cd \ ab	00	01	11	10
00	0	1	1	0
01	1	1	×	0
11	1	0	0	0
10	0	0	0	0