Col215P SW1

Manshi Sagar, 2020CS50438
Richa Yadav, 2020CS50438

28 October 2022

1 Implementation

In the **comb-function-expansion** function, we first make a k-map from the True and Don’t Care terms using the gray-code ordering. We used a helper function **make-gray-code** to compute gray code ordering(separately for columns and rows) as done in assignment 1. Then, we called the max-legal-region function for all terms in the func-TRUE array.

The helper function **max-legal-region** takes as input : k map matrix, term array and returns as output : maximally reduced term.
In this function, we check all possible legal regions by replacing some variables with None value.
First we change one variable to None and check if the newly formed term has a legal region in the matrix or not.
If the term has a legal region, then we add this term to a new expansion array.
Now, we repeat the process of trying to find the expanded term by replacing more variables with None.
After completion of each step, in the expansion array, all terms have equal number of None terms and this number keeps increasing after each iteration.
Function terminates when the expansion array becomes empty and returns the last element added to the expansion array in the previous iteration.

2 Testcases

```
func_TRUE = ["a’b’", "ab’"]
func_DC = ["ab"]
FINAL EXPANDED TERMS: ["b’", ’a’]

func_TRUE = ["a’b’c’d"]
func_DC = ["abc’d", "a’b’c’d’", "a’bc’d’", "abc’d’", "ab’c’d’", "a’bc’d", "a’b’cd",
"a’bcd", ’abcd’, "a’bcd’", "abcd’"]
FINAL EXPANDED TERMS: ["a’c’"]

func_TRUE = ["a’b’c’"]
func_DC = ["ab’c’", "a’b’c", "ab’c", "a’bc"]
FINAL EXPANDED TERMS: ["b’"]

func_TRUE = ["a’b’c’d’e’", "a’b’cd’e", "a’b’cde’", "a’bc’d’e’", "a’bc’d’e",
"a’bc’de", "a’bc’de’", "ab’c’d’e’", "ab’cd’e’"]
func_DC = ["abc’d’e’", "abc’d’e", "abc’de", "abc’de’"]
FINAL EXPANDED TERMS: ["c’d’e’", "a’b’cd’e", "a’b’cde’", "bc’", "bc’", "bc’", "bc’",
"c’d’e’", "ab’d’e’"]

func_TRUE = ["a’b’c’d", "a’b’cd", "a’bc’d’"]
func_DC = ["abc’d’", "ab’c’d", "ab’cd", "a’bcd’", "abcd’"]
FINAL EXPANDED TERMS: ["b’d", "b’d", "bd’"]
```

```
func_TRUE = ["a’b’c’d", "a’bc’d", "abc’d’", "ab’cd’"]
func_DC = ["a’bc’d’", "a’b’cd", "a’bcd", "abcd’"]
FINAL EXPANDED TERMS: ["a’d", "a’d", "abd’", "acd’"]

func_TRUE = ["a’b’c’d’", "a’bc’d’", "a’bc’d"]
func_DC = ["abc’d’", "ab’c’d’", "abc’d", "a’bcd", ’abcd’, "a’bcd’", "abcd’"]
FINAL EXPANDED TERMS: ["c’d’", ’b’, ’b’]

func_TRUE = ["a’b’c’d’e’", "a’bcd’e’", "a’b’c’de", "a’b’cde", "a’bc’d’e"]
func_DC = ["a’bc’d’e’", "ab’c’d’e’", "a’bcd’e", "a’bc’de", "ab’c’de", "a’bc’de’"]
FINAL EXPANDED TERMS: ["a’c’d’e’", "a’bd’", "a’b’de", "a’b’de", "a’bc’"]

func_TRUE = ["a’b’c’d"]
func_DC = ["abc’d", "a’b’c’d’", "a’bc’d’", "abc’d’", "ab’c’d’", "a’bc’d", "a’b’cd",
"a’bcd", "abcd", "a’bcd’", "abcd’"]
FINAL EXPANDED TERMS: ["a’c’"]

func_TRUE = ["a’bcde", "a’b’c’d’e’", "a’bcd’e’", "a’b’c’de", "a’b’cde", "a’bc’d’e"]
func_DC = ["a’bc’d’e’", "ab’c’d’e’", "a’bcd’e", "a’bc’de", "ab’c’de", "a’bc’de’"]
FINAL EXPANDED TERMS: ["a’be", "a’c’d’e’", "a’bd’", "a’de", "a’de", "a’bc’"]

func_TRUE = ["a’bcde", "a’bcd’e’", "a’b’c’de", "a’b’cde", "a’bc’d’e"]
func_DC = ["a’bc’d’e’", "ab’c’d’e’", "a’bcd’e", "a’bc’de", "ab’c’de", "a’bc’de’",
"abc’de", "abcde", "ab’cde"]
FINAL EXPANDED TERMS: [’de’, "a’bd’", ’de’, ’de’, "a’bc’"]
```