COL 215 SW Assignment 2

Rajat Bhardwaj
2020CS50436
Geetansh Juneja
2020CS50649

October 2022

1 Report

We first found a set that represents all the legal regions. We did that as follows. We inserted a term(from True + dont care) in a set, before inserting we checked whether there exists a term in the set already which can combine with this term to double the region. If yes we remove both terms from the set and add the double region term in the list so that it can be checked and added in the future. If there doesnot exist any term in the set which can help in doubling the area we simple insert that term in the set.
Basically we found a string with one change in the set (example for "abcd'e"' if "ab'cd'e"' is present in the set already , we remove both "abcd'e"' and "ab'cd'e"' and we add "acd'e"' in a list which has to be checked in future.

We added the set to an array and we sorted all the elements of the array such that the elements with less literals lies before the elements with more literals. Now for each term in the True region, we iterate in the array, if we find that the term lies in the region of the element of the array, we return this region (this region will be maximal since the array is sorted in increasing size of the literals i.e. decreasing size of the region) and break the iteration.
Hence we find all the maximal regions.

2 Functions

2.1 convert_list_to_string (L)

This function converts a list of string to one string by concatenation every element

2.2 convert_string_to_list(s : string)

This function converts a string of literals to a list

2.3 doubleregion(s)

It takes a term as input. This region returns an array of tuple. Each tuple has 2 elements. Firstly the region which is double region of s and the complementary region such that the complementary region + s = double region. It returns all such possible regions

2.4 insert(regionset , F , legal_regions)

This function builds legal_regions list. It takes into input the terms which needs to be added and using double region, it finds all possible double regions and checks whether the complementary term(as explained above) lies in the region set already or not. It appends F (excluding the terms for which we found a double region) to legal_regions list

2.5 fallin(term , reg)

This function checks if the term lies in the legal region reg or not.

2.6 comb_function_expansion(func_TRUE, func_DC)

This function calls insert() function to build the legal_regions list, Then for each term of func_True it checks the legal_regions to find the corresponding maximal legal region of that term.
We have hardcoded the case when the maximal legal region is the entire k-map. We have also hardcoded the case when func_True is empty. All other cases shall be handled by the code.

3 Testcases

3.1 Test Case 1

func_TRUE = "abc'd" , "abcd" , "a'b'c'd" , "a'b'c'd'" func_DC = "a'bc'd", "abc'd'" output = ["abc'", "a'b'c'", 'abd', "a'b'c'"]

3.2 Test Case 2

func_TRUE = "abc'de" , "abcde'" , "a'b'c'de'" , "a'b'c'd'e","abcde","a'bc'd'e" func_DC = "a'bc'de", "abc'd'e", "a'bc'de'", "abc'd'e'" output = ["bc'e", 'abcd', "a'c'de'", "bc'e", "a'c'd'e", 'abde']

3.3 Test Case 3

func_TRUE = "abc'def" , "abcde'f'" , "a'b'c'de'f'" , "a'b'c'd'ef","abcdef'","a'bc'd'ef'" , "abcd'e'f", "abcdef"
func_DC = "a'bc'def", "abc'd'ef", "a'bc'de'f", "abc'd'e'f" , "abcdef'"

output = ['abdef', "abcdf"", "bc'def", "abd'e'f", "abcdf"", "a'bc'd'ef"", "a'b'c'de'f"", "a'b'c'd'ef"]

3.4 Test Case 3

func_TRUE = "abc'defghi" , "abcde'f'ghi" , "a'b'c'de'f'ghi" , "a'b'c'd'efghi","abcdef'ghi"", "a'bc'd'ef'ghi"" ,"abcd'e'fghi"", "abcdefghi"", "ab'c'd'e'fghi", "abc'defg'hi""
func_DC = "a'bc'defghi"", "abc'd'efghi"", "a'bc'de'fghi", "abc'd'e'fghi" , "abcdef'ghi"
output = ["a'b'c'de'f'ghi", "ac'd'e'fghi", "a'b'c'd'efghi", "abc'defghi", "abcdeghi"", "abc'defg'hi"", "abcdef'gh", "a'bc'd'ef'ghi"", "abcd'e'fghi"", "abcdf'ghi"]

4 Do all expansions result in an identical set of terms?

No, all expansions do not result in an identical set of terms. Because there may be more than one regions of same and maximal size for a particular term. Basically we expand a term and we get more than one regions for expansions, we keep the regions of the largest size and delete the rest. Therefore there may be more than one region of largest size hence all expansions do not result in identical set of terms.

5 Are all expansions equally good, assuming that our objective is to maximally expand each term? Explain

No, all expansions are not equally good. The expansions having the minimum number of literals are good whereas he expansions having more literals are not good. There can be multiple expansions of a term. We will keep the expansion which maximum region size (least number of literals)