SW Assignment-2

Atul Jeph 2020CS10329 Aryan Gaurav 2020CS10327

October 2022

1 Approach

Precisely, our approach was to first make a kmap of appropriate size using the true and don't care values provided, then for each true term we remove one variable from the term and check if the new term is legal or not, if yes then we expand the new expanded term further and if no then we put back the variable and remove the next, and finally putting the term in the final output list and at the end returning this output list.

2 Answers to the questions

1. Do all expansions result in an identical set of terms?

Ans. No, there can be more than one expansion for a term with the same max size.

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

Ans. Yes, all expansions are equally good as size of region they carry are all equal i.e. the maximum size possible for that particular term's expansion.

3 Code

The **Code** consists 7 functions **region**(term,reg), **is_legal_region**(kmap_function, term), **expand_term**(kmap_function, term, reg, index), **str_to_term**(stri, no_of_variables), **term_to_str**(final_term), **all_combinations**(final_term, index, combs), **comb_function_expansion**(func_TRUE, func_DC) here the first 2 are directly used from the last submission, the rest are explained below:-

1. **expand_term**(kmap_function, term, reg, index)

   This functions job is to expand a term as max as possible by removing most of the variables from the term alongside maintaining its legality.

This is done by checking all 2^n cases of keeping a variable or not keeping this variable, and finding max region size that it could attain and returning the corresponding minimized term.

Then using dynamic programing over it to improve runtimes to 2-3 times faster.

2. **str_to_term**(stri, no_of_variables)

   This functions has a simple job of converting the string terms into a list of size = no. of variables where i^{th} element represent state of (a+i) characters value. For ex. If we have abc' then output[0] will be 1, if a'bc' then 0 and if bc' then None.

3. **term_to_str**(final_term)

   This functions is complete opposite of **str_to_term** function.

   So its job is just to convert that list representation of the term back in string format.

4. **all_combinations**(final_term, index, combs)

   This function is mainly for the demo purpose only, all it dose is find the difference in region of final expanded term and initial terms and returns it in a list format.

5. **comb_function_expansion**(func_TRUE, func_DC)

   This is our main function which creates a kmap as a 2d list using the func_TRUE and func_DC provided and then it uses the obove functions and mainly the **expand_term** function to find the expanded terms corresponding to each true term on the kmap and finally return all as a list of strings.

4 Test Cases

Input:

func_TRUE = ["abcdef'g'h'", "a'bcdefgh", "a'bc'd'e'f'gh", "a'b'cd'ef'g'h'", "abc'de'f'gh", "a'b'c'd'ef'g'h", "a'bc'd'efgh", "a'bc'defgh", "abcd'e'fg'h", "a'b'cd'e'fgh'", "a'b'cde'f'g'h'", "a'bc'd'efg'h", "a'b'cd'e'fgh", "ab'cde'f'gh'", "ab'cd'e'f'gh'", "ab'cd'e'fg'h", "a'bc'd'e'fg'h'", 'abcdefgh', "a'b'c'd'efgh", "a'b'cdef'g'h'", "a'b'c'def'g'h'", "abc'de'f'g'h", "abc'd'e'f'g'h'", "abcd'e'fgh'", "ab'cd'e'fgh'", "abc'd'efg'h'", "a'bcdefg'h", "a'bc'de'fgh'", "ab'cd'ef'gh'", "ab'c'de'f'g'h", "abc'de'fg'h", "a'bcdef'gh'", "abc'd'ef'gh'", "ab'c'defgh", "abcd'e'f'gh'", "ab'cd'efg'h", "ab'cde'fg'h'", "a'bc'de'fg'h", "a'b'cde'fgh'", "a'bcde'fg'h'", "abc'd'e'f'g'h", "a'bcdef'gh", "a'b'c'de'f'g'h'", "ab'c'd'efgh", "a'b'cdefgh", "a'bcd'e'f'g'h'", "ab'c'd'ef'gh", "a'bcd'e'f'g'h", "a'b'c'def'g'h", "a'b'cd'e'fg'h", "ab'cde'f'g'h", "a'b'c'def'gh'", "ab'c'de'fg'h'", "abcdefg'h", "abcd'ef'gh'", "a'b'cd'ef'g'h", "a'b'c'de'fg'h'", "a'b'c'd'ef'g'h'", "ab'c'def'g'h'", "a'b'c'd'ef'gh", "abc'def'g'h'", "abcdef'gh", "a'b'c'd'e'fgh'", "a'bc'def'gh", "abcde'fg'h", "a'b'cdefgh'", "ab'cdefgh", "a'bcde'f'gh", "ab'c'def'gh", "ab'cdefg'h", "a'bc'de'f'g'h", "abc'defg'h", "ab'c'de'fg'h", "ab'c'd'e'f'gh", "ab'c'd'e'fg'h"], ["abc'd'ef'", "a'b'c'de'f'", "a'b'cde'f'", "abcde'f", "ab'c'd'ef'", "a'bc'd'ef'", "abcde'f'", "abcd'e'f", "abc'def'", "abcd'ef'", "a'bcd'e'f"], ["abc'd'efg", "abcd'e'f'g", "abc'de'fg'", "a'bcde'fg'", "a'bcde'f'g", "abc'd'ef'g'", "abcd'e'fg'", "ab'cde'fg", "abcdef'g'", "a'b'c'de'fg'", "ab'c'd'e'f'g'", "a'b'cd'ef'g'", "a'b'cde'fg", "a'bcd'e'f'g", "abcd'ef'g'", "a'bc'de'f'g", "ab'c'de'fg'", "a'b'c'def'g", "ab'c'd'efg", "ab'c'd'e'fg", "ab'c'd'ef'g'", "a'b'c'd'e'fg", "a'bc'd'e'fg", "ab'cd'e'fg'", 'abcdefg', "abcdefg'", "abcd'e'fg", "a'bcdef'g", "a'bc'd'efg'", "a'b'cdefg", "a'bcd'efg'", "a'b'cdefg'", "ab'c'd'ef'g"], ["a'b'c'd'e'f'g'h", "a'b'cde'fg'h'", "ab'cd'efgh", "a'bc'de'fgh'", "a'bcdefgh", "abc'defg'h'", "a'b'cdefgh'", "ab'cd'e'f'gh", "ab'c'd'efg'h'", "ab'c'd'e'fgh", "a'bc'def'gh", "a'b'c'd'efg'h'", "a'b'c'd'ef'gh", "abcdef'gh'", "a'b'cd'ef'g'h'", "a'b'c'defgh'", "a'b'cd'e'fg'h", "ab'cde'f'g'h", "a'b'c'def'gh'", "ab'c'de'fg'h'", "abcdefg'h", "a'b'cd'ef'g'h", "a'b'c'd'ef'g'h'", "ab'c'def'g'h'", "a'b'c'd'ef'gh", "abc'def'g'h'", "abcdef'gh", "a'b'c'd'e'fgh'", "a'bc'def'gh", "abcde'fg'h", "a'b'cdefgh'", "ab'cdefgh", "a'bcde'f'gh", "ab'c'def'gh", "ab'cdefg'h", "a'bc'de'f'g'h", "abc'defg'h", "ab'c'de'fg'h", "ab'c'd'e'f'gh", "ab'c'd'e'fg'h"], ["abc'd'efg", "abcd'e'f'g", "abc'de'fg'", "a'bcde'fg'", "a'bcde'f'g", "abc'd'ef'g'", "abcd'e'fg'", "ab'cde'fg", "abcdef'g'", "a'b'c'de'fg'", "ab'c'd'e'f'g'", "a'b'cd'ef'g'", "a'b'cde'fg", "a'bcd'e'f'g", "abcd'ef'g'", "a'bc'de'f'g", "ab'c'de'fg'", "a'b'c'def'g", "ab'c'd'efg", "ab'c'd'e'fg", "ab'c'd'ef'g'", "a'b'c'd'e'fg", "a'bc'd'e'fg", "ab'cd'e'fg'", 'abcdefg', "abcdefg'", "abcd'e'fg", "a'bcdef'g", "a'bc'd'efg'", "a'b'cdefg", "a'bcd'efg'", "a'b'cdefg'", "ab'c'd'ef'g"], ["a'b'c'd'e'f'g'h", "a'b'cde'fg'h'", "ab'cd'efgh", "a'bc'de'fgh'", "a'bcdefgh", "abc'defg'h'", "a'b'cdefgh'", "ab'cd'e'f'gh", "ab'c'd'efg'h'", "ab'c'd'e'fgh", "a'bc'def'gh", "a'b'c'd'efg'h'", "a'b'c'd'ef'gh", "abcdef'gh'", "a'b'cd'ef'g'h'", "a'b'c'defgh'", "a'b'cd'e'fg'h", "ab'cde'f'g'h", "a'b'c'def'gh'", "ab'c'de'fg'h'", "abcdefg'h", "abcd'ef'gh'", "a'b'cd'ef'g'h", "a'b'c'de'fg'h'", "a'b'c'd'ef'g'h'", "ab'c'def'g'h'", "a'b'c'd'ef'gh", "abc'def'g'h'", "abcdef'gh", "a'b'c'd'e'fgh'", "a'bc'def'gh", "abcde'fg'h", "a'b'cdefgh'", "ab'cdefgh", "a'bcde'f'gh", "ab'c'def'gh", "ab'cdefg'h", "a'bc'de'f'g'h", "abc'defg'h", "ab'c'de'fg'h", "ab'c'd'e'f'gh", "ab'c'd'e'fg'h"], ["abc'd'ef'", "a'b'c'de'f'", "a'b'cde'f'", "abcde'f", "ab'c'd'ef'", "a'bc'd'ef'", "abcde'f'", "abcd'e'f", "abc'def'", "abcd'ef'", "a'bcd'e'f"], ["abc'd'efg", "abcd'e'f'g", "abc'de'fg'", "a'bcde'fg'", "a'bcde'f'g", "abc'd'ef'g'", "abcd'e'fg'", "ab'cde'fg", "abcdef'g'", "a'b'c'de'fg'", "ab'c'd'e'f'g'", "a'b'cd'ef'g'", "a'b'cde'fg", "a'bcd'e'f'g", "abcd'ef'g'", "a'bc'de'f'g", "ab'c'de'fg'", "a'b'c'def'g", "ab'c'd'efg", "ab'c'd'e'fg", "ab'c'd'ef'g'", "a'b'c'd'e'fg", "a'bc'd'e'fg", "ab'cd'e'fg'", 'abcdefg', "abcdefg'", "abcd'e'fg", "a'bcdef'g", "a'bc'd'efg'", "a'b'cdefg", "a'bcd'efg'", "a'b'cdefg'", "ab'c'd'ef'g"], ["a'b'c'd'e'f'g'h", "a'b'cde'fg'h'", "ab'cd'efgh", "a'bc'de'fgh'", "a'bcdefgh", "abc'defg'h'", "a'b'cdefgh'", "ab'cd'e'f'gh", "ab'c'd'efg'h'", "ab'c'd'e'fgh", "a'bc'def'gh", "a'b'c'd'efg'h'", "a'b'c'd'ef'gh", "abcdef'gh'", "a'b'cd'ef'g'h'", "a'b'c'defgh'", "a'b'cd'e'fg'h", "ab'cde'f'g'h", "a'b'c'def'gh'", "ab'c'de'fg'h'", "abcdefg'h", "abcd'ef'gh'", "a'b'cd'ef'g'h", "a'b'c'de'fg'h'", "a'b'c'd'ef'g'h'", "ab'c'def'g'h'", "a'b'c'd'ef'gh", "abc'def'g'h'", "abcdef'gh", "a'b'c'd'e'fgh'", "a'bc'def'gh", "abcde'fg'h", "a'b'cdefgh'", "ab'cdefgh", "a'bcde'f'gh", "ab'c'def'gh", "ab'cdefg'h", "a'bc'de'f'g'h", "abc'defg'h", "ab'c'de'fg'h", "ab'c'd'e'f'gh", "ab'c'd'e'fg'h"], ["a'b'cd'ef", "a'bc'def"], ["a'b'cde"], ["abcd'e'", "a'b'c'de'", "a'bcde", "abc'de", "ab'c'de'", "a'bcd'e", "a'bc'd'e", "ab'cd'e'", "a'b'c'd'e", "ab'cde", "abc'd'e"], ["a'b'cd'efg'", "ab'c'd'efg", "ab'c'de'fg'", "ab'cdefg'"], ["abcde'fgh'", "a'b'c'd'e'fg'h", "a'bcde'fgh'", "a'bcde'f'g'h'", "a'b'c'd'ef'gh'", "a'b'c'def'g'h", "a'b'c'd'ef'gh", "a'bcd'ef'gh'", "abcdef'gh", "a'bc'd'e'f'g'h'", "ab'cde'fg'h'", "a'b'c'd'efgh'", "a'bcd'ef'g'h'", "a'bc'd'e'fg'h", "abc'de'fg'h'", "a'bc'd'e'fg'h'", "a'bc'defgh", "abcd'e'f'gh", "abc'de'fgh", "abcd'efgh", "ab'cde'f'g'h'", "a'bcde'fgh", "abcde'fg'h'"], ["a'bcd'efg'h'i", "ab'c'd'e'fghi", "ab'cde'f'g'h'i'", "a'b'c'de'fg'h'i'", "a'bc'de'fg'h'i'", "abcdefghi'", "a'b'c'defghi'", "a'b'cde'f'g'h'i", "abcdef'gh'i", "abcd'efgh'i", "ab'cd'efgh'i'", "ab'cd'ef'gh'i", "a'bcdefg'h'i'", "ab'c'de'fg'hi'", "a'bc'def'g'hi",

"abcde'fg'h'i'", "a'bc'de'fgh'i'", "abc'de'f'g'hi", "ab'c'd'e'fg'h'i'", "ab'c'de'fgh'i'",
"a'bcde'f'g'h'i'", "ab'c'd'ef'g'hi'", "a'bc'd'e'fg'hi'", "a'bcd'e'fg'hi", "a'bcde'fg'hi'",
"ab'c'd'efg'h'i'", "a'bc'de'fghi'", "a'b'c'def'ghi", "a'bcdef'g'hi", "abc'de'f'ghi",
"a'b'cde'fgh'i", "a'bcdef'g'h'i'", "abcde'f'g'h'i'", "abc'de'fg'h'i'", "a'b'cd'ef'g'h'i'",
"a'bcdef'gh'i'", "ab'c'de'fghi", "abc'd'e'f'g'hi'", "abc'd'e'fg'h'i", "abcde'f'g'hi",
"a'b'cdefgh'i'", "abc'de'fg'hi'", "a'b'cde'fghi", "a'b'c'd'e'fg'hi'", "a'bc'd'ef'gh'i'",
"a'b'c'd'ef'g'h'i'", "a'b'cd'e'fg'h'i", "a'b'c'd'e'f'ghi", "abc'd'efg'h'i", "abc'def'g'h'i'",
"a'b'c'd'e'f'g'hi'", "a'bcde'fg'hi", "a'b'c'd'efg'hi'", "a'b'c'd'efg'h'i'", "abc'defg'hi'"]

func_DC = ["a'bcdefg'h'", "abcde'fgh'", "a'b'cde'fgh", "a'b'cdef'gh", "a'b'cde'f'gh'",
"a'bc'd'e'f'g'h", "ab'cdefg'h'", "a'bc'de'fgh", "abcd'efg'h", "abcdefgh'", "ab'c'd'ef'gh'",
"ab'c'defg'h'", "abcdef'gh'", "abcd'efgh'", "a'bc'de'f'gh", "a'bc'd'ef'gh'", "a'b'cde'fg'h",
"a'b'c'd'e'f'gh", "a'bcd'e'fgh", "ab'c'de'f'gh", "abcd'efgh", "a'b'cdef'gh'", "a'b'cde'f'g'h",
"abc'de'fgh", "a'bcd'ef'gh", "abc'd'e'fg'h'", "ab'c'd'e'fg'h'", "a'bc'def'gh'", "a'b'c'd'efg'h",
"ab'c'defgh'", "abc'def'gh", "a'bc'de'f'g'h'", "a'bc'def'g'h", "abc'd'ef'gh", "abcdef'g'h",
"a'bcd'ef'g'h'", "abc'defgh'", "a'b'c'defg'h'", "a'b'c'de'fg'h", "a'b'c'd'e'f'g'h'",
"a'bc'd'ef'g'h", "a'b'cd'efg'h'", "abc'de'f'g'h'", "a'bcdefgh'", "a'bc'd'efg'h'", "abc'defg'h'",
"a'bcde'fgh", "a'b'c'd'e'f'g'h", "a'b'cd'efgh'", "a'b'c'de'fgh'", "a'b'cd'e'fg'h'",
"a'b'cd'ef'gh'", "ab'cd'ef'g'h", "a'bcd'e'fgh'", "ab'c'd'ef'g'h", "a'b'c'de'fgh", "abc'd'efgh'",
"ab'cdef'g'h", "a'bc'd'e'f'g'h'", "a'b'cdefg'h", "abcde'f'gh", "ab'cd'e'fg'h'", "a'bc'defg'h",
"a'b'cdef'g'h", "a'bc'd'ef'g'h'", "a'b'c'de'f'gh", "a'bc'd'e'fgh'", "a'b'c'd'e'f'gh'",
"abcde'f'g'h'", "a'b'cd'efg'h", "ab'cd'ef'gh", "abc'def'gh'", "a'b'c'd'e'fg'h'", "abc'd'e'fgh'",
"ab'c'defg'h", "abcde'f'gh'", "abc'defgh", "abcd'e'f'g'h'", "a'bcd'e'fg'h", "a'bc'd'ef'gh",
"ab'c'de'f'gh'", "a'bcdef'g'h", "ab'cd'ef'g'h'", "a'b'c'defg'h", "a'b'c'd'e'fg'h",
"abc'd'ef'g'h'"], ["a'b'cd'ef'", "a'bc'de'f", "ab'cd'ef", "a'bcdef", "a'bc'def'", "ab'cde'f'",
"abcd'ef", "abc'de'f'", "abcdef'", "a'b'c'd'e'f", "abc'de'f", "a'bcde'f"], ["a'b'cdef'g",
"a'b'c'd'e'f'g'", "a'b'cd'e'fg", "abcde'f'g", "abcde'fg", "ab'cde'f'g", "ab'cd'efg'", "a'bcdefg",
"ab'cde'fg'", "a'b'c'd'efg", "a'bc'defg'", "ab'cde'f'g'", "a'bcd'e'fg", "ab'c'd'e'fg'",
"abc'd'e'fg", "abc'defg'", "abcd'ef'g", "a'bcd'ef'g'", "ab'c'de'fg", "ab'c'd'efg'", "a'bcd'e'fg'",
"a'bcd'efg", "a'bc'de'fg'", "abc'defg", "a'b'cde'fg'", "abc'd'e'f'g", "abc'de'f'g", "ab'c'def'g'",
"a'bc'def'g", "a'bc'd'e'fg'", "a'b'cd'efg'", "a'bcd'e'f'g'"], ["a'bcde'fg'h", "abcde'fgh'",
"a'b'c'def'g'h'", "abcd'e'f'gh'", "abcd'ef'g'h'", "ab'cdef'g'h'", "a'bcd'e'f'gh", "a'bcd'e'fgh",
"a'bc'def'g'h", "a'b'c'd'e'f'gh", "a'b'cd'ef'g'h", "ab'cd'e'fg'h'", "abcdefgh'", "ab'c'de'f'g'h",
"abcd'e'fg'h", "abc'd'efg'h", "ab'cde'f'gh", "a'b'c'de'fg'h'", "a'b'c'd'ef'gh'", "ab'c'd'e'fg'h'",
"a'b'c'de'fgh", "a'b'cdef'gh'", "ab'cde'f'gh'", "abcd'ef'gh'", "ab'cde'fg'h", "ab'cd'e'fgh",
"abcd'efg'h'", "ab'c'defgh", "a'bc'd'e'fg'h", "a'bcde'f'g'h", "ab'cd'efg'h", "abc'defgh",
"ab'cd'efgh'", "ab'c'def'g'h'", "a'b'c'def'gh'", "a'bc'd'efg'h", "a'bcd'e'f'g'h'", "abc'd'e'fg'h'",
"a'b'cde'f'gh'", "abcd'e'f'gh", "a'bcd'efgh'", "a'b'c'de'fg'h", "ab'c'def'gh", "a'b'c'd'e'f'g'h'",
"a'b'cd'e'f'g'h", "a'bcd'ef'gh'", "abcdef'g'h'", "ab'cdefg'h'", "a'bcde'f'gh'", "ab'c'd'efgh",
"abcdefg'h", "abc'd'e'f'gh'", "ab'c'd'ef'gh'", "a'bcdefg'h'", "a'bc'd'e'fg'h'", "ab'c'd'e'f'gh'",
"a'b'cd'e'fgh'", "abc'de'fg'h'", "ab'c'defgh'", "a'b'cd'efg'h", "a'bc'd'efgh", "ab'cd'ef'g'h",
"a'bc'defgh'", "ab'c'de'fg'h'", "ab'c'd'ef'g'h'", "ab'cd'ef'gh", "a'bc'd'efgh'", "abc'd'ef'gh",
"a'b'c'de'f'g'h'", "abc'defg'h", "a'b'c'd'ef'g'h", "ab'cdefgh", "ab'cde'fgh", "abcdef'g'h",
"a'b'cd'efgh", "a'bcdefg'h", "abcd'e'f'g'h", "a'b'cd'ef'gh", "a'bc'de'f'gh"], ['abcdef',
"abc'def'", "a'bcd'e'f'", "ab'c'de'f", "abcd'ef'", "abcde'f", "ab'cd'e'f'"], ["a'b'c'de'",
"a'bc'd'e'", "a'bc'd'e"], ["abcde'", "ab'c'd'e'", "abc'd'e'", "abcd'e", "a'bcde'", "ab'c'de",
"a'bcd'e'", "ab'cd'e", "a'b'cd'e'", "a'b'c'de", "a'b'cd'e"], ["ab'c'de'f'g", "a'bc'd'ef'g"],
["abc'd'e'f'g'h'", "a'bcdef'g'h", "a'bc'de'f'g'h'", "a'bcd'efg'h", "abc'd'ef'g'h'", "a'b'cde'fgh",
"a'b'c'defg'h", "ab'c'd'efg'h'", "ab'c'd'e'fgh'", "abcd'e'f'g'h'", "ab'cdefgh'", "a'bcde'f'g'h",

"a'bc'd'ef'gh'", "a'bcde'f'gh'", "ab'cd'e'f'g'h", "a'bc'defg'h", "a'bc'd'e'fgh'", "ab'c'de'fg'h", "a'bc'd'e'f'gh'", "abc'd'e'fg'h", "a'bc'defg'h'", "ab'cd'efgh", "abc'defg'h'"], ["abcd'ef'gh'i'", "ab'c'de'f'ghi'", "a'b'c'd'e'fgh'i'", "a'bc'd'e'fg'h'i'", "abcd'e'fg'hi'", "a'bc'defg'hi'", "a'b'cde'f'g'hi'", "abc'd'e'fg'h'i'", "abc'defghi", "a'bc'd'e'f'g'h'i'", "abc'd'e'fghi", "ab'c'd'ef'g'h'i", "abcd'ef'g'h'i", "ab'c'd'e'f'ghi", "ab'cde'f'g'hi", "abc'de'f'ghi'", "ab'c'de'f'gh'i'", "a'b'cd'e'f'g'hi", "a'b'c'de'fgh'i'", "ab'c'd'e'f'g'hi", "a'bc'd'efgh'i'", "a'b'c'd'ef'ghi'", "abcde'fghi'", "a'bc'd'ef'g'h'i", "a'bcd'e'fghi'", "ab'cde'f'gh'i'", "a'bc'def'gh'i", "a'bcd'e'f'g'h'i'", "a'bcdefg'h'i", "a'bcd'e'f'g'hi'", "a'b'c'd'e'fgh'i", "a'bc'de'fg'hi'", "ab'c'defgh'i", "a'bcde'f'g'hi'", "a'bc'd'e'f'g'hi'", "a'b'c'd'e'fg'h'i", "a'bc'de'f'ghi", "a'b'c'd'e'fghi", "abc'd'efgh'i", "abc'd'ef'g'h'i", "a'bcd'efg'h'i'"]

outputs:

["abcdef'", "a'bdeh", "a'bc'f'h", "a'b'ef'g'", "bc'de'h", "a'b'ef'g'", "a'bc'eh", "a'bc'dh", "abcfg'h", "a'b'cd'e'f", "a'b'cdf'g'", "a'bc'eh", "a'b'cd'e'f", "ace'f'gh'", "acd'e'gh'", "b'd'e'fg'", "bc'd'g'h'", "abdeg', "a'c'd'eh", "a'b'ef'g'", "a'b'ef'g'", "bc'de'h", "bc'd'g'h'", "abcgh'", "acd'e'gh'", "abc'eh'", "a'bdeh", "a'c'de'fg", "ab'cd'ef'", "ac'de'f'h", "bc'de'h", "a'cdeg", "abc'eh'", "ac'def", "abcgh'", "b'ceg'h", "ab'dfg'h'", "a'bc'dh", "a'b'cdfg", "a'bcdfg'h'", "bc'e'f'g'", "a'bdeh", "a'b'c'dg'h'", "ab'c'egh", "a'cdeg", "a'bd'e'f'g'", "ab'c'egh", "a'bd'e'f'g'", "a'b'ef'g'", "a'b'fg'h", "b'cdf'g'h", "a'b'def'h'", "b'c'dfg'", 'adefh', "abcgh'", "a'b'ef'g'", "b'c'dfg'", "a'b'ef'g'", "ac'deg'h'", "a'c'd'eh", "abc'eh'", 'abdeg', "a'b'c'd'e'h'", "a'bc'dh", "abcfg'h", "a'cdeg", 'adefh', "a'bdgh", "ab'c'egh", 'adefh', "a'bc'dh", "ac'def", "b'c'dfg'", "ab'c'f'gh", "b'c'e'fg'"]

Passed 0

["abef'", "a'b'de'f'", "a'b'de'f'", "abde'", "ac'd'ef'", "bc'ef'", "abde'", "bce'f", "abdf'", "abef'", "bce'f"]

Passed 1

["ac'd'fg", "abce'g", "bc'dfg'", "a'be'fg'", "a'bdf'g", "abd'ef'g'", "bcd'e'f", "ab'cde'", "abcdeg'", "a'de'fg'", "ab'c'd'g'", "a'cd'eg'", "a'b'cdf", "a'bcd'e'", "abcd'ef'",

"a'bdf'g", "ab'c'e'f", "a'def'g", "ab'c'd'e", "ab'c'd'f", "ab'c'd'e", "a'd'e'fg", "a'bd'e'f", "ab'd'fg'", 'abdef', 'abdef', "abce'g", "a'bdf'g", "a'bc'fg'", "a'b'cdf", "a'bcd'f", "a'b'cdf", "ab'c'd'e"]

Passed 2

["a'b'd'f'g'", "a'b'cde'g'", "ab'cd'h", "bc'dfh'", "a'bdefh", "bc'def", "b'cefg", "ab'cd'h", "ab'c'd'eh'", "ab'd'fgh", "a'bc'de", "b'c'egh'", "a'b'c'd'ef'", "abcdef'", "a'b'd'f'g'", "a'c'deh'", "abc'de'f'", "bc'def", "ab'cd'ef", "a'c'dg'h'", "a'bc'de", "a'b'ce'g'h'", "a'ce'f'g'", "acd'e'h", "abc'd'gh", "a'bc'fg'", "ab'ce'f'g", "ab'c'def'", "ab'ce'fg'", "b'cefg", "bc'dfh'", "a'bc'de", "a'befg'", "ab'efg", "a'b'cde'g'", "a'ce'f'g'", "ab'efg", "a'ce'f'g'", "a'bfg'h", "abcd'e'f'", "a'bc'de", "abcdef'", "a'cdf'gh'", "abc'de'f'", "b'c'ef'h'", "b'cfgh", "b'cfgh", "a'bc'fg'", "ab'cd'h", "abcd'e'g'", "a'b'd'f'g'", "a'b'd'f'g'", "a'c'deh'", "abc'd'gh", "a'ce'f'g'", "a'bcd'ef'g", "b'cefg", "a'bc'ef", "a'bc'de", "ab'cd'h", "a'bc'de", "abc'de'g", "a'bc'd'fh", "a'befg'", "a'c'e'fg'h", "abc'de'f'", "bc'efh", "bc'dg'h'", "bc'dg'h'", "c'de'g'h'", "ab'c'df'h"]

Passed 3

["a'b'cd'ef", "a'bc'def"]

Passed 4

["a'b'cde"]

Passed 5

["cd'", "b'c'd", "a'bc", "abc'e", "b'c'd", "cd'", "a'd'e", "cd'", "a'd'e", "ab'ce", "abd'"]

Passed 6

["a'b'cd'efg'", "ab'c'd'efg", "ab'c'de'fg'", "ab'cdefg'"]

Passed 7

["abcde'fh'", "a'c'd'e'fg'h", "a'bcde'gh'", "a'bcde'f'", "a'b'c'd'ef'g", "a'b'c'deg'h",
"a'b'c'd'ef'g", "a'bcd'ef'h'", "abcdef'gh", "a'bc'd'e'h'", "ab'cde'g'h'", "a'b'c'd'egh'",
"a'bcd'ef'h'", "a'bc'd'e'fg'", "abc'dfg'h'", "a'bc'd'e'h'", "a'bc'defh", "abcd'e'f'gh",
"abc'de'fgh", "acd'efgh", "ab'cde'g'h'", "a'bcde'f'", "abcde'fh'"]

Passed 8

["a'bcefg'h'", "b'c'd'e'ghi", "ab'cde'f'h'i'", "a'c'de'fh'i'", "a'bc'de'fi'", "abcdfghi'",
"a'b'c'defghi'", "a'b'cde'f'g'h'i", "abcdef'gh'i", "abd'efgh'i", "ab'cd'efgh'i'", "ab'cd'ef'gh'i",
"a'bcefg'h'", "ac'de'fg'hi'", "a'bdef'g'hi", "abcde'g'h'i'", "a'bc'de'fi'", "abc'de'f'hi",
"ab'c'd'fg'h'i'", "ab'c'de'gh'i'", "a'bce'f'g'i'", "ab'c'd'ef'g'hi'", "a'bc'd'e'g'i'", "a'bce'fg'hi",
"a'bcde'fg'h", "ab'c'd'fg'h'i'", "a'bc'de'fi'", "a'b'c'def'ghi", "a'bdef'g'hi", "abc'de'f'gh",
"a'b'cde'fgi", "a'bcdef'h'i'", "abcde'g'h'i'", "bc'de'fg'i'", "a'b'd'ef'g'h'i'", "a'bcdef'h'i'",
"ab'c'e'fghi", "bc'd'e'f'g'hi'", "abc'd'e'fg'h'", "abde'f'g'hi", "a'b'cdefgh'i'", "bc'de'fg'i'",
"a'b'cde'fgi", "a'c'd'e'g'hi'", "a'bc'd'egh'i'", "a'b'c'd'eg'h'i'", "a'b'd'e'fg'h'i", "b'c'd'e'ghi",
"abc'd'efh'i", "abc'def'g'h'i'", "a'c'd'e'g'hi'", "a'bcde'fg'h", "a'b'c'd'efg'i'", "a'b'c'd'efg'i'",
"bc'dfg'hi'"]

Passed 9