Quine-McCluskey

Quine-McCluskey method is an algorithm that minimizes Boolean expressions **without limiting the number of input variables** **unlike the Karnaugh map.** The automation of the Quine-McCluskey algorithm is also quite easy to do.

[Term project] Implement the Quine-McCluskey algorithm in C/C++ for a Boolean function with any number of inputs larger than zero. The input and output file formats are shown below.

Input format	**Output format**
File name: input_minterm.txt	File name: result.txt
4 // input bit length d 0000 // don't care value m 0100 // input having the result with true m 0101 m 0110 m 1001 m 1010 d 0111 d 1101 d 1111	01-- 1-01 1010 Cost (# of transistors): 40

Cost of a SOP expression

Q-M algorithm computes a SOP expression of a function. A SOP expression can be implemented by a two-level logic circuit composed of inverters, AND gates, and an OR gate. Let the *cost* of a SOP expression be the *number of transistors* needed for the two-level logic circuit implementation of the SOP.

For example, A'BC' + C'DEF needs two inverters (4 transistors), one three-input AND gate (8 tr.), one four-input AND gate (10 tr.), and one two-input OR gate (6 tr.). The cost of A'BC' + C'DEF is 28.

What you need to submit:

1. Report
   - A. Problem statement (15%)
   - B. Your algorithm with pseudo code and flow chart (30%)
   - C. Verification strategy & corresponding examples with explanation (50%)
   - D. A testcase that you think it is very hard to solve (5%)
     - ✓ This might be used to verify the results of classmates.
2. Program source code with sufficient comments (No English comment allowed except any terminologies)
   - ✓ You can use any C++ standard libraries, but no external libraries (like Boost C++) are allowed.
   - ✓ Your program must be single-threaded. No multi-threading is allowed.
3. Project file except .vs folder (Visual studio version: above 2019)
4. Examples that you used to verify your program (The quantity as well as the quality of examples are very important).

5. **Bonus**

   A. **Optimal cost winners (10%):** There can be multiple SOP expressions of a function. If you implement an algorithm to find the *minimum cost* SOP expression, you will get extra 10% points**.** Up to *seven* (7) winners will get the bonus points based on the average SOP cost of the test cases provided by the lecturer.

   B. **Fastest algorithm winners (10%):** Among the up to seven (7) optimal cost winners, the top *five* (5) fastest implementations of the Q-M algorithm will get extra bonus 10%, 8%, 6%, 4%, and 2% points accordingly.

   C. **Memory-efficient algorithm winners (10%):** Among the optimal cost winners, the top five memory-efficient (i.e., low memory usage) implementations of the Q-M algorithm will get extra bonus ~10% points accordingly. Your memory usage will be measured using our program below URL: https://github.com/ESALKW/QM-Memory-Check
   (Note that we will not respond any questions about memory usage tracking program.)

   D. Note that only one execution file for a team will be used for the bonus credit. Please plan your strategy wisely.

6. **Due date:** 2023. 04. 21. (Fri) 23:59
   Late submission (due by 2023. 04. 22 (Sat) 23:59) will decrease 20% of your project score.

Grade will be given based on the following criteria.

1. Reports: 40%

2. Program source & binary code & examples: 20%

   ✓ Insufficient comment, memory leakage, I/O error, or any program error occurred will decrease your program score. So be careful.

3. Test with our examples: 40%

4. Extra points (when accomplished bonus problem): Up to +30%

   ✓ You will be a bonus candidate only when your program has no error and pass our test.

5. If you cheat, such as using ChatGPT, copying and pasting your classmate's result, the assignment will be 0 points. **You should make your program yourself!!**

How to submit your project

Report and program: Compress your report and program into single zip file and upload it to KLAS assignment menu. The filename of your zip file should be “학번_Project.zip”. (e.g., 2022112233_Project.zip)

Have a question about this project?

Ask your teaching assistant about this project.

TA information:

- E-mail: kdsh5800+2023qm@gmail.com
- Date: from 2022/04/07 to 2022/04/20 23:59

Note that if your question is about algorithm or implementation itself, your TA may not able to reply the question.