

HW1 Quine-McCluskey Method

Author: Yuanchung Lee(李元中)

Sturdent ID: 3125110522

My C++ Code Overview:

There are 6 file in the 312511052.tar, the description is below:

1. **312511052.cpp:**

- This C++ program is to perform logic minimization using the Quine-McCluskey and Petrick methods, which is the main cpp file.
- It reads input data from a file, including the number of variables, the "OnSet," and the "Don't Care" set.
- It uses the Quine-McCluskey method to generate prime implicants and the Petrick method to find the optimal Sum-of-Products (SOP) representation.
- The program outputs the results to a file, including prime implicants and the minimized expression in alphabetical notation.

2. **classtype.h:**

- This header file defines several structures and classes used in the logic minimization program, including **QuineMcCluskey**, **Petrick**, **COLUMN**, and **ROW**.
- These structures and classes are used for representing and manipulating the data required for logic minimization.

3. **QuineMcCluskey:**

- The **QuineMcCluskey** class implements the Quine-McCluskey method for generating prime implicants.
- It includes functions for converting integers to binary strings, sorting binary strings, and merging implicants.
- The main function **GeneratePrimImplicants** generates prime implicants, which will be return as a vector of strings.

4. **COLUMN:**

- The **COLUMN** class represents a column in the Quine-McCluskey method.
- It includes functions for converting integers to binary strings, checking if it is a subset of prime implicants, and appending prime implicants.

5. **ROW:**

- The **ROW** class represents a row in the Quine-McCluskey method, storing prime implicants and columns that cover the OnSet terms.
- The constructor calculates the number of literals in the prime implicant.

6. Petrick:

- The **Petrick** class implements the Petrick method for logic minimization, both in the Product-of-Sums (POS) and Sum-of-Products (SOP) forms.
- It includes functions for generating the POS form, finding the optimal SOP representation, and performing recursive exploration for minimal SOP expressions to find the best solution.

Functionality:

- The program reads input data describing a Boolean function, performs logic minimization, and outputs the minimized expression.
- The Quine-McCluskey method is used to generate prime implicants, and the Petrick method is used to find an optimal SOP expression.

Contribution:

- The provided code performs important tasks in logic minimization, which is crucial for optimizing digital circuits and simplifying Boolean expressions.
- The Quine-McCluskey and Petrick methods are implemented effectively for generating prime implicants and finding minimal SOP expressions.

How to Use

1. **Compile the code** using a C++ compiler.

```
make
```

2. **Run the program** with the following command:

```
./312511052.o input_file output_file
```