Self Reflection

Group 24 Lucy Cui

Introduction

Our group has implemented a tool to simplify propositional expressions with the help of Karnaugh map (Kmap). This reflection will contain our brainstorming and implementing process, shortcomings of the tool, my personal contribution and gain, and how the tool would change if a different target group was selected.

Brainstorm and Implementation

In the process of studying other modules, our group members agreed that propositional logic was the hardest, especially when it came to simplify propositional expressions to their simplest Disjunctive Normal Forms (DNF). We decided to focus on this topic and Peter introduced the Kmap.

When designing coding structure, we divided the work into two main parts, algorithms and the user interface. After coding for each part, we arranged meetings to build connections and integrate the codes.

Shortcomings

We succeeded in completing almost all the functions we planned in our scenario 1 report. However, there are still points to improve. The most important thing that we need to do next is to beautify our user interface. Currently, the user interface is crude, and it only contains boxes to input and buttons to press. More effort and time is needed to make it look better. At the same time, the current user interface shows slightly different colors on different devices, for example, Mac and Windows. We also need to take time to unify that.

Another point we need to consider is to optimize the code and make it more readable. With the time limit, we only managed to finish the code, so optimization is needed to improve its speed, and we need to tidy the code or adding comments so that other people can understand it easily.

Contribution and Gain

I was responsible for the algorithm part of the code. I learned the concept of Kmap and implemented an simplifying algorithm for our project. The input of the algorithm is the binary representations of all combinations of valuations which would make the complicated propositional expression true. The output is the binary representations of the simplified expression. The algorithm works by combining terms with only 1 bit different, and using Cartesian product to get the smallest set of sum of products.

After finishing my part, I met with Alexio, who was in charge of the user interface part. To integrate two parts of code, I implemented another function to convert the binary representation into string representation, so that the result could be shown directly.

From the project, I learned how Kmap works, and practiced object oriented programming in Python. I also learned some new data structures in Python, including default dict and frozen set. During group work, I improved my communication skills and collaborated with my team members well. I familiarized myself with Github since we used it for version control and collaboration. I knew some new friends and had fun working with them.

Target Audience

The target group we selected was high school and college students who have a certain foundation on propositional logic and wish to improve simplification skills. If our target audience is middle school students who have no or little foundation, the design of the project would change dramatically. Instead of focusing on using Kmap for simplification, we would pay more attention to simpler topics, such as building a truth table step by step. We may provide a propositional expression and a certain valuation, and ask the student whether the valuation will make the expression true or not. A summary of rules for each connective can be shown on the side to help the students.

Conclusion

In conclusion, working on this project was a valuable and meaningful experience for me. I improved my programming and communication skills as well as having fun working with my teammates. Although parts of our project can be further improved, I believe this experience will be helpful and beneficial for my future projects.