

Obtaining Reduced Expression of Boolean Functions

Group Members -

Devraj Priyadarshi 19EC30012

Azim Sohel 19EE30003

Piyush Kumar 19IE10022

Problem Statement :

Problem Description

Obtaining Reduced form of SOP(Sum of Product) / POS(Product of Sum) form of a function from its given truth table i.e. writing boolean expressions in minimized form. We take minterms/maxterms/Karnaugh maps as inputs, use searching methods to reach to the output with optimized boolean expression.

Mathematical Interpretation

Every boolean function can be expressed as a sum of minterms or a product of maxterms. Since the number of literals in such an expression is usually high, and the complexity of the digital logic gates that implement a Boolean function is directly related to the complexity of the algebraic expression from which the function is implemented, it is preferable to have the most simplified form of the algebraic expression. The process of simplifying the algebraic expression of a boolean function is called minimization.

Inputs :- The Truth table of the boolean function. This can be of the form of minterms or maxterms.

Output :- A reduced form of the same boolean function.

The process of minimization is based upon several boolean laws, but in the case of functions with too many variables, we might get overwhelmed applying all those rules. Thus we approach this problem with a search based procedure, explained below.

AI Mapping & Approach

State Space Searching. The action may be taken out by forming minterm/maxterms using a Karnaugh map as an input and the result is the number of terms in the output . A goal state can be where the number of terms is coming minimum. There may be many goal states to one problem, we'll consider one of them as our final output.

State Space : Every variable considered in the expression and their combinations.

State : [combination of variables in SOP or POS form]

Goal State : the state where the expression gives the same result as our input function for all values of variables.

Heuristics & Cost : based on addition of each combination, with bigger expressions having bigger cost.

Programming Language - Python.

FILES : <https://drive.google.com/drive/folders/1TkHB6-MwtntOt8aR6hth9b-E6Nq3DeOz?usp=sharing>

