Este trabajo es una búsqueda de implementación de un método de minimización de funciones booleanas, vista desde un punto teórico y llevado a uno práctico; haciendo para esta segunda parte mencionada un programa hecho en lenguaje JavaScript. Sin embargo, antes de profundizar de lleno al método que nos compete en esta ocasión, mencionemos un poco de los antecedentes que nos llevaron a la creación de este trabajo.

Para la materia Estructuras Discretas, impartida por el profesor Orlando Zaldívar Zamorategui, hemos abordado el manejo de la lógica proposicional, como una forma de demostración de la validez de ciertos razonamientos de interés. Si bien, en el lenguaje común los podemos encontrar en forma de presentación de un evento y efectuar una categorización de la información dada, como proposiciones que forman una premisa (o hipótesis) y llegar a una conclusión (que no siempre es válida); siendo este caso un método inductivo. O una oración en la cual se nos describe una situación general que va a la particular; un ejemplo podría ser: “Todos los perros ladran.”, y “Lanudo es un perro, por lo tanto ladra.”; de esto hay miles de ejemplos, mismos que son métodos de deductividad, método que a nosotros como parte del curso de Estructuras Discretas nos compete y trabajaremos de una forma distinta a lo que nos ofrecen los anteriores ejemplos.

Bien, de estas proposiciones, que deben ser entendidas como enunciados declarativos que pueden adquirir el valor de verdadero o falso; al momento de reproducirlas en dos categorías: Atómicas/primarias/simples y Compuestas/moleculares/secundarias; estas segundas derivadas de las atómicas, pues las atómicas representan enunciados declarativos (que solo presentan información tomada como verdadera, podríamos decir); son capaces de poseer conectivos (unarios o binarios). Quedando para nosotros trabajar con una simbología estricta, siendo las representaciones de las proposiciones letras mayúsculas (aquí se debe ser claro, solamente las atómicas pueden ser representadas por letras mayúsculas), y sus conectivos serán negación (representada por 7), disyunción (representada por $\vee$) conjunción (representada por $\wedge$), condicional (representada por $\rightarrow$), y por último el bicondicional (representado por $\leftrightarrow$). De aquí se desglosa todo un tópico que denominado cálculo de predicados, que deriva en otros asuntos que refieren a su propio manejo como lo dictan sus leyes y propiedades, que son regidas por la parte más básica de toda la lógica proposicional que son las tablas de verdad que ayudan a verificar a su vez la validez de las leyes mencionadas.

Aunque, si bien este es un tema de arduo interés. Ya podemos ir mencionando que, de aquí, de la formación de las fórmulas proposicionales que se crean usando atómicas y compuestas que muestran una presunta validez, que en lenguaje de lógica proposicional se representa a la equivalencia con el símbolo ⇔. Para el algebra booleano, es distinto en cuanto a simbología se refiere y el uso de los últimos dos conectivos mencionados, la condicional y la bicondicional; además de que la simbología de representación de lo que conocíamos como atómicas, de mayúsculas pasan a ser minúsculas. Y en resumen, tenemos que:

- Negación (7) ‘
- Disyunción (˅) +
- Conjunción (˄) ·
- Y para la equivalencia (⇔) =

Claro, lo anterior son ejemplos de expresiones booleanas simples. Pero, precisamente, por los casos en que se presentan expresiones complejas y extensas es que se necesitan aplicar ciertos métodos de minimización para obtener la expresión más simple. Y ese es el objetivo de este proyecto que se resumen en: investigar, comprender e implementar de forma teórica y por medio de un software el método de minimización de QuineMcCluskey para funciones booleanas.

Por lo que, para llegar a cumplirlo, se dividió el proyecto en varias secciones. La primera sección, se encargará de proporcionar las definiciones y conceptos del algebra booleana contando con ejemplos y apoyándonos una vez más de la tabla VI. Leyes y propiedades (algebra booleana); abordándolo desde la base que tenemos en la aritmética, el algebra común que conocemos para realizar operaciones matemáticas; para seguir con el algebra booleana que hemos mencionado brevemente en esta introducción (pero con ejemplos y aplicando las leyes para buscar minimizar sin ayuda de los métodos de minimización); mencionar los métodos de minimización existentes, para dar introducción al método de QuineMcCluskey.

Para la segunda sección, vamos a proporcionar ejemplos de la aplicación del método QuineMcCluskey en forma de imágenes y explicación por medio de notas agregadas para una mejor interpretación por parte del usuario. Seguido de un vídeo abordando la parte teórica necesaria, y un segundo para realizar un ejemplo. Y, como recomendación al usuario: repase el trabajo, pues este proyecto está acompañado de un breve cuestionario que le ayudará a repasar el tema. Si aún hay dudas, o quiere un apoyo para un problema más complejo: se adjunta un programa hecho con tal de proporcionar una solución al usted, el usuario.