Scientific Research Publishing

Journal of Quantum Information Science, 2026, 16(1), 120-131
https://www.scirp.org/journal/jqis
ISSN Online: 2162-576X
ISSN Print: 2162-5751

Quantum Algorithm for Solving Tautology and Satisfiability Problems

Amrita Mitra

Independent Researcher, Bangalore, India
Email: mitra.amrita2017@gmail.com

How to cite this paper: Mitra, A. (2026) Quantum Algorithm for Solving Tautology and Satisfiability Problems. *Journal of Quantum Information Science*, **16**, 120-131.
https://doi.org/10.4236/jqis.2026.161004

Received: November 16, 2025
Accepted: March 10, 2026
Published: March 13, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

This paper proposes a quantum algorithm for solving the tautology and the satisfiability problems for a Boolean formula. Let's say we are given a Boolean formula. The variables of the Boolean formula can take only two values—TRUE or FALSE. The tautology problem asks whether the Boolean formula always evaluates to TRUE for all values of its Boolean variables. The satisfiability problem asks whether the Boolean formula can evaluate to TRUE for at least one set of values of its Boolean variables. This paper proposes that both problems can be solved using the proposed quantum algorithm. The tautology problem is known to be co-NP-complete. The satisfiability problem is known to be NP-complete.

Keywords

Quantum Algorithm, Tautology Problem, Satisfiability Problem, NP-Completeness, Co-NP-Completeness

1. Introduction

Let's say we are given a Boolean formula with n number of Boolean variables—$x_0, x_1, \cdots, x_{n-1}$. Each of these n variables can take two possible values: TRUE or FALSE. Therefore, n variables can take 2^n different values in total. If the Boolean formula is a tautology, then the Boolean formula will always evaluate to TRUE, regardless of the values of the variables.

The tautology problem is co-NP-complete [1]. As we know, a problem is in co-NP if its complement is in NP. For example, the complement of the tautology problem asks whether the Boolean formula evaluates to TRUE for at least one combination of values of the Boolean variables. Please note that this is the same problem as the satisfiability problem.

The satisfiability problem is in NP. If the variables of the Boolean formula are

assigned Boolean values, we can verify in polynomial time whether the Boolean formula evaluates to TRUE.

Moreover, the satisfiability problem is known to be NP-complete. As the satisfiability problem is the complement of the tautology problem, the tautology problem is a co-NP-complete problem.

A qubit does not always take the values 0 or 1. It can take any values of the form $\left(\alpha|0\rangle+\beta|1\rangle\right)$, where α and β are two complex numbers and $|\alpha|^2+|\beta|^2=1$. We can leverage this to solve the tautology and the satisfiability problem using a quantum algorithm.

2. The Proposed Quantum Algorithm

Let's say $\mathcal{F}$ is a Boolean formula consisting of n input variables. The input variables are Boolean and can take the values TRUE or FALSE. Depending on these n Boolean variables, $\mathcal{F}$ can evaluate to any value, TRUE or FALSE.

Let's also assume that $f:\{0,1\}^n\to\{0,1\}$ is a function. The n input Boolean variables of the function $f:\{0,1\}^n\to\{0,1\}$ represent the n Boolean variables of $\mathcal{F}$. And the output of f represents the output of the Boolean formula $\mathcal{F}$.

In other words, let's say $x_0,x_1,\cdots,x_{n-1}$ are the n input variables of $f:\{0,1\}^n\to\{0,1\}$. $f\left(x_{n-1},x_{n-2},\cdots,x_0\right)$ represents the value that the Boolean formula $\mathcal{F}$ evaluates for the same set of values of its Boolean variables. Let's also assume that the function $f:\{0,1\}^n\to\{0,1\}$ can be implemented as an oracle.

Therefore, if $\mathcal{F}$ is a tautology, then $f:\{0,1\}^n\to\{0,1\}$ always evaluates to 1. If $\mathcal{F}$ is not satisfiable, then $f:\{0,1\}^n\to\{0,1\}$ always evaluates to 0. In all other cases, if $f:\{0,1\}^n\to\{0,1\}$ evaluates to 1 for at least one set of values of the input variables, then $\mathcal{F}$ is satisfiable.

For our purpose, we can use the quantum circuit as shown in Figure 1. The first n qubits are initialized to $|0\rangle^{\otimes n}$, and the last qubit is initialized to $|0\rangle$.

We can apply the Hadamard gate on the first n qubits. As a result, after applying the Hadamard gate, we will obtain the following state:

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle \tag{1}$$

Now, we can apply the oracle [2] U_f that implements the function f. Therefore, if we query the oracle, we obtain the following state:

$$|\psi\rangle=\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\left|f\left(x\right)\right\rangle \tag{2}$$

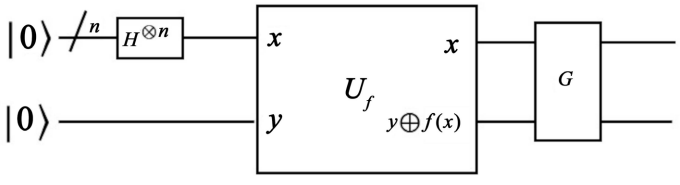

Figure 1. Quantum circuit implementing the proposed quantum algorithm.

For example, if $n=2$, we will get the following state:

$$\begin{aligned}|\psi\rangle &= \frac{1}{\sqrt{2^2}}\sum_{x=0}^{3}|x\rangle|f(x)\rangle \\ &= \frac{|00\rangle|f(00)\rangle+|01\rangle|f(01)\rangle+|10\rangle|f(10)\rangle+|11\rangle|f(11)\rangle}{2}\end{aligned} \tag{3}$$

At this point, if $\mathcal{F}$ is a tautology, we will get the following state:

$$|\psi\rangle = \frac{1}{\sqrt{2^2}}\sum_{x=0}^{3}|x\rangle|f(x)\rangle = \frac{|001\rangle+|011\rangle+|101\rangle+|111\rangle}{2} \tag{4}$$

If $\mathcal{F}$ is not satisfiable, we will get the following state:

$$|\psi\rangle = \frac{1}{\sqrt{2^2}}\sum_{x=0}^{3}|x\rangle|f(x)\rangle = \frac{|000\rangle+|010\rangle+|100\rangle+|110\rangle}{2} \tag{5}$$

However, if $\mathcal{F}$ is satisfiable, it can be in any other state, like the following, for example:

$$|\psi\rangle = \frac{1}{\sqrt{2^2}}\sum_{x=0}^{3}|x\rangle|f(x)\rangle = \frac{|000\rangle+|011\rangle+|100\rangle+|110\rangle}{2} \tag{6}$$

The $(n+1)$ qubits of the quantum state $|\psi\rangle$ now passes through the quantum gate G. We will use the unitary matrix M to implement G. The construction of the $2N\times 2N$ unitary matrix M is explained in Section 3.

The output of G is $(n+1)$ qubit. The measurements of the output qubits specify whether $\mathcal{F}$ is satisfiable, unsatisfiable, or a tautology.

3. Construction of the Unitary Matrix *M*

The quantum gate G is implemented using the unitary matrix M. n is the number of Boolean variables of $\mathcal{F}$ and $N=2^n$.

Let's consider the quantum state $|\psi\rangle$.

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle = a_0|x_0\rangle+a_1|x_1\rangle+\cdots+a_{2N-1}|x_{2N-1}\rangle \tag{7}$$

Please note that $|x_i\rangle$ for $0\le i\le 2N-1$ has $(n+1)$ qubits. a_i and a_{i+1} cannot be non-zero simultaneously, as the output of $f(x)$ for a specific x cannot be both 0 and 1. For example, if $n=2$ and $f(00)=0$, then the coefficient a_0 for $|000\rangle$ is $\frac{1}{\sqrt{N}}$ and the coefficient a_1 for $|001\rangle$ is 0.

Let's say the output of the quantum gate G is the following:

$$b_0|x_0\rangle+b_1|x_1\rangle+\cdots+b_{2N-1}|x_{2N-1}\rangle \tag{8}$$

Therefore, we can write the following:

$$M\begin{bmatrix}a_0\\a_1\\\vdots\\a_{2N-1}\end{bmatrix} = \begin{bmatrix}b_0\\b_1\\\vdots\\b_{2N-1}\end{bmatrix} \tag{9}$$

In our case, M is a $2N \times 2N$ unitary matrix. Let's say M is the following block matrix:

$$M = \frac{1}{\sqrt{2}} \begin{bmatrix} A & A \\ A & -A \end{bmatrix} \tag{10}$$

As M is a unitary and real matrix, we can write the following:

$$MM^{\mathrm{T}} = I$$

$$\text{or, } \frac{1}{2} \begin{bmatrix} A & A \\ A & -A \end{bmatrix} \begin{bmatrix} A & A \\ A & -A \end{bmatrix} = I$$

$$\text{or, } \frac{1}{2} \begin{bmatrix} 2A^2 & 0 \\ 0 & 2A^2 \end{bmatrix} = I$$

$$\therefore A^2 = I \tag{11}$$

We will consider the following initial value for A:

$$A_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{12}$$

We observe that

$$A_0^{\,2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \tag{13}$$

We can now put the value of A_0 and construct M_0 as follows:

$$M_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} A_0 & A_0 \\ A_0 & -A_0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \tag{14}$$

We can now set $A_1 = M_0$ and calculate M_1 as $M_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} A_1 & A_1 \\ A_1 & -A_1 \end{bmatrix}$.

Similarly, we can continue to construct the $2^{n+1} \times 2^{n+1}$ matrix $M = M_{n-1}$ where $N = 2^n$. We will discuss the time complexity of constructing matrix M in Section 5.

4. Some Properties of the Matrix *M*

Property 1: The first and second rows of $M_{2^{n+1} \times 2^{n+1}}$ will always contain 0 and $\frac{1}{\sqrt{2^n}}$ alternately.

We can use mathematical induction to prove this.

$$M_0 = M_{2^2 \times 2^2} = \frac{1}{\sqrt{2}} \begin{bmatrix} A_0 & A_0 \\ A_0 & -A_0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \tag{15}$$

M_0 contains 0 and $\frac{1}{\sqrt{2}}$ alternately in the first and second rows. Therefore,

the property holds for M_0. Let's say the property holds for all M_k for $0 \le k \le n$. We will prove that the property holds for M_{k+1}.

$$M_{k+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} A_{k+1} & A_{k+1} \\ A_{k+1} & -A_{k+1} \end{bmatrix} \tag{16}$$

$$A_{k+1} = M_k \tag{17}$$

As per our assumption, the property holds for $M_k = M_{2^{k+2} \times 2^{k+2}}$. As the number of rows of $A_{k+1} = M_k$ for $k \ge 0$ is more than 2, the first and second rows of $M_{k+1} = M_{2^{k+3} \times 2^{k+3}}$ will contain $\frac{1}{\sqrt{2^{k+1}}} \times \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{2^{k+2}}}$ and 0 alternately. Hence, the property holds for all $k \ge 0$.

Property 2: Each row i of $M_{2^{n+1} \times 2^{n+1}}$, for $2 \le i < 2^{n+1} - 1$, contains $\frac{1}{\sqrt{2^n}}$ and $\frac{-1}{\sqrt{2^n}}$ alternately as non-zero elements.

We can use mathematical induction to prove this.

$$M_0 = M_{2^2 \times 2^2} = \frac{1}{\sqrt{2}} \begin{bmatrix} A_0 & A_0 \\ A_0 & -A_0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}_{2^2 \times 2^2} \tag{18}$$

Each row i for $2 \le i \le 2^2 - 1$ contains $\frac{1}{\sqrt{2}}$ and $\frac{-1}{\sqrt{2}}$ alternately as non-zero elements. Therefore, the property holds for M_0.

Let's say the property holds for all M_k for $0 \le k \le n$. We will prove that the property also holds for M_{k+1}.

$$M_{k+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} A_{k+1} & A_{k+1} \\ A_{k+1} & -A_{k+1} \end{bmatrix} \tag{19}$$

$$A_{k+1} = M_k \tag{20}$$

As the property holds for M_k, it will hold for A_{k+1}. And the non-zero elements of $M_{k+1} = M_{2^{k+3} \times 2^{k+3}}$ will be $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2^{k+1}}} = \frac{1}{\sqrt{2^{k+2}}}$ and $\frac{-1}{\sqrt{2^{k+2}}}$ alternately. Therefore, the property holds for M_{k+1}. Hence, the property holds for all $k \ge 0$.

Property 3: The matrix M is unitary.

M is constructed recursively from A_0. Therefore, the following holds:

$$M_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} A_0 & A_0 \\ A_0 & -A_0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}_{2^2 \times 2^2}$$

$$M_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} A_1 & A_1 \\ A_1 & -A_1 \end{bmatrix} = \frac{1}{\sqrt{2^2}} \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & \vdots & & \\ & \vdots & & \end{bmatrix}_{2^3 \times 2^3}$$

$$M_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} A_2 & A_2 \\ A_2 & -A_2 \end{bmatrix} = \frac{1}{\sqrt{2^3}}\begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & & \vdots & \\ & & \vdots & \end{bmatrix}_{2^4\times 2^4}$$

$$\vdots$$

$$M_{n-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} A_{n-1} & A_{n-1} \\ A_{n-1} & -A_{n-1} \end{bmatrix} = \frac{1}{\sqrt{2^n}}\begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & & \vdots & \\ & & \vdots & \end{bmatrix}_{2^{n+1}\times 2^{n+1}} \tag{21}$$

We can use mathematical induction to prove that A_k is an involutory [3] matrix for $k \geq 0$.

$$A_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$A_0 A_0^{\mathrm{T}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = I \tag{22}$$

As A_0 is a symmetric matrix, $A_0 = A_0^{\mathrm{T}}$. Therefore,

$$A_0^2 = A_0 A_0^{\mathrm{T}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = I \tag{23}$$

Therefore, A_k is an involutory matrix for $k = 0$. Let's assume that A_k is an involutory matrix for all k where $0 \leq k \leq n-1$.

$$\begin{aligned} A_k = M_{k-1} &= \frac{1}{\sqrt{2}}\begin{bmatrix} A_{k-1} & A_{k-1} \\ A_{k-1} & -A_{k-1} \end{bmatrix} \\ \therefore A_k^2 = A_k A_k^{\mathrm{T}} &= \frac{1}{\sqrt{2}}\begin{bmatrix} A_{k-1} & A_{k-1} \\ A_{k-1} & -A_{k-1} \end{bmatrix} \times \frac{1}{\sqrt{2}}\begin{bmatrix} A_{k-1} & A_{k-1} \\ A_{k-1} & -A_{k-1} \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} A_{k-1} & A_{k-1} \\ A_{k-1} & -A_{k-1} \end{bmatrix}^2 = \frac{1}{2}\begin{bmatrix} 2A_{k-1}^2 & 0 \\ 0 & 2A_{k-1}^2 \end{bmatrix} = \begin{bmatrix} A_{k-1}^2 & 0 \\ 0 & A_{k-1}^2 \end{bmatrix} \end{aligned} \tag{24}$$

As per our assumption, A_{k-1} is involutory.

$$\therefore A_k^2 = \begin{bmatrix} A_{k-1}^2 & 0 \\ 0 & A_{k-1}^2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{25}$$

Therefore, A_k is an involutory matrix for all k where $0 \leq k \leq n-1$.

As $M_k = \frac{1}{\sqrt{2}}\begin{bmatrix} A_k & A_k \\ A_k & -A_k \end{bmatrix}$, we can write the following:

$$\begin{aligned} M_k M_k^{\mathrm{T}} &= \frac{1}{\sqrt{2}}\begin{bmatrix} A_k & A_k \\ A_k & -A_k \end{bmatrix} \times \frac{1}{\sqrt{2}}\begin{bmatrix} A_k & A_k \\ A_k & -A_k \end{bmatrix}^{\mathrm{T}} \\ &= \frac{1}{2}\begin{bmatrix} A_k & A_k \\ A_k & -A_k \end{bmatrix}^2 = \frac{1}{2}\begin{bmatrix} 2A_k^2 & 0 \\ 0 & 2A_k^2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \tag{26}$$

Hence, $M_{n-1}=\frac{1}{\sqrt{2}}\begin{bmatrix}A_{n-1} & A_{n-1}\\ A_{n-1} & -A_{n-1}\end{bmatrix}=\frac{1}{\sqrt{2^n}}\begin{bmatrix}0 & 1 & 0 & \cdots\\ 1 & 0 & 1 & \cdots\\ & \vdots & & \\ & \vdots & & \end{bmatrix}_{2^{n+1}\times 2^{n+1}}$ is a unitary matrix.

5. Output of the Quantum Gate G

As discussed in Section 3,

$$|\psi\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle=a_0|x_0\rangle+a_1|x_1\rangle+\cdots+a_{2N-1}|x_{2N-1}\rangle \tag{27}$$

And the output of the quantum gate G is the following:

$$b_0|x_0\rangle+b_1|x_1\rangle+\cdots+b_{2N-1}|x_{2N-1}\rangle \tag{28}$$

Therefore, we can write the following:

$$M\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_{2N-1}\end{bmatrix}=\begin{bmatrix}b_0\\ b_1\\ \vdots\\ b_{2N-1}\end{bmatrix} \tag{29}$$

As discussed in Section 4, the first two rows of M alternate between 0 and 1. Therefore, we can write the following:

$$\frac{1}{\sqrt{N}}\begin{bmatrix}0 & 1 & 0 & \cdots\\ 1 & 0 & 1 & \cdots\\ & \vdots & & \\ & \vdots & & \end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_{2N-1}\end{bmatrix}=\begin{bmatrix}b_0\\ b_1\\ \vdots\\ b_{2N-1}\end{bmatrix}$$

$$b_0=\frac{1}{\sqrt{N}}(a_1+a_3+a_5+\cdots)$$

$$b_1=\frac{1}{\sqrt{N}}(a_0+a_2+a_4+\cdots) \tag{30}$$

The input of the quantum gate G is the following:

$$|\psi\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle=a_0|x_0\rangle+a_1|x_1\rangle+\cdots+a_{2N-1}|x_{2N-1}\rangle \tag{31}$$

The first n qubits of $|x_k\rangle$ for $0\le k\le 2N-1$ and $N=2^n$ specify the input of the given Boolean function and the $(n+1)$ th qubit of $|x_k\rangle$ is the output of the Boolean function. Therefore, a_i and a_{i+1} cannot be 1 together for $0\le i\le 2N-2$.

If the given Boolean function is a tautology, the $(n+1)$ th qubit of $|x_k\rangle$ for $0\le k\le 2N-1$ and $N=2^n$ is always 1. Therefore, in that case, $a_0,a_2,a_4,\cdots$ are zeros and $a_1,a_3,a_5,\cdots$ are all ones.

$$\therefore b_0=\frac{1}{\sqrt{N}}(a_1+a_3+a_5+\ldots)=1$$

$$\therefore b_1=\frac{1}{\sqrt{N}}(a_0+a_2+a_4+\ldots)=0$$

$$\therefore b_i = 0 \text{ for } 2 \le i \le 2N-1 \tag{32}$$

Please note that $b_i = 0$ for $2 \le i \le 2N-1$ as per Property 2. Therefore, the output of the quantum gate G, in that case, will be the following:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Similarly, if the given Boolean function is not satisfiable, the $(n+1)$ th qubit of $|x_k\rangle$ for $0 \le k \le 2N-1$ and $N = 2^n$ is always 0. Therefore, in that case, $a_0, a_2, a_4, \cdots$ are all ones and $a_1, a_3, a_5, \ldots$ are all zeros. Therefore, in that case,

$$b_0 = \frac{1}{\sqrt{N}}\left(a_1 + a_3 + a_5 + \cdots\right) = 0 \tag{33}$$

$$b_1 = \frac{1}{\sqrt{N}}\left(a_0 + a_2 + a_4 + \cdots\right) = 1 \tag{34}$$

$$b_i = 0 \text{ for } 2 \le i \le 2N-1 \tag{35}$$

Therefore, the output of the quantum gate G, in that case, will be the following:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

In all other cases, the output of the quantum gate G will be different from $|0\cdots00\rangle$ or $|0\cdots01\rangle$. The Boolean function, in that case, is satisfiable. We can detect that by measuring the $(n+1)$ output qubits.

If we repeat the experiment, the $(n+1)$ output qubits, in that case, will collapse to different values each time. The $(n+1)$ output qubits, in that case, will not be $|0\cdots00\rangle$ or $|0\cdots01\rangle$.

6. Time Complexity for Constructing *M*

M is constructed recursively from A_0.

$$M_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} A_0 & A_0 \\ A_0 & -A_0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}_{2^2 \times 2^2}$$

$$M_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} A_1 & A_1 \\ A_1 & -A_1 \end{bmatrix} = \frac{1}{\sqrt{2^2}}\begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & & \vdots & \\ & & \vdots & \end{bmatrix}_{2^3 \times 2^3}$$

$$M_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} A_2 & A_2 \\ A_2 & -A_2 \end{bmatrix} = \frac{1}{\sqrt{2^3}}\begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & & \vdots & \\ & & \vdots & \end{bmatrix}_{2^4\times 2^4}$$

$$\vdots$$

$$M_{n-2} = \frac{1}{\sqrt{2}}\begin{bmatrix} A_{n-2} & A_{n-2} \\ A_{n-2} & -A_{n-2} \end{bmatrix} = \frac{1}{\sqrt{2^{n-1}}}\begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ & & \vdots & \\ & & \vdots & \end{bmatrix}_{2^n\times 2^n} \tag{36}$$

In the ith iteration, we compute M_i. Therefore, we need $O(n)$ iterations to compute $M_{n-1} = M_{2^{n+1}\times 2^{n+1}}$, where n is the number of variables of the Boolean function $\mathcal{F}$.

7. Discussion on a Specific Corner Case

Let's consider a scenario where the Boolean function has n variables. And it evaluates to 1 in only one specific assignment of Boolean values to the set of n variables.

$$N = 2^n$$

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle = a_0|x_0\rangle + a_1|x_1\rangle + \cdots + a_{2N-1}|x_{2N-1}\rangle$$

$$M\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2N-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2N-1} \end{bmatrix}$$

$$\therefore b_0 = \frac{1}{\sqrt{N}}(a_1 + a_3 + a_5 + \cdots) = \frac{1}{N}$$

$$b_1 = \frac{1}{\sqrt{N}}(a_0 + a_2 + a_4 + \cdots) = \frac{(N-1)\frac{1}{\sqrt{N}}}{\sqrt{N}} = 1 - \frac{1}{N}$$

$$|b_i| = \frac{1}{N} \text{ for } 2 \le i \le 2N-1 \tag{37}$$

As discussed earlier, if the given Boolean function is not satisfiable, $a_0, a_2, a_4, \cdots$ are all ones and $a_1, a_3, a_5, \cdots$ are all zeros. Therefore, in that case,

$$b_0 = \frac{1}{\sqrt{N}}(a_1 + a_3 + a_5 + \cdots) = 0$$

$$b_1 = \frac{1}{\sqrt{N}}(a_0 + a_2 + a_4 + \cdots) = 1$$

$$b_i = 0 \text{ for } 2 \le i \le 2N-1 \tag{38}$$

Therefore, the probability of incorrectly getting $|00\cdots01\rangle$ instead of $|00\cdots00\rangle$ is significantly high.

To address this problem, we can use Unambiguous State Discrimination [4] [5].

Let's say the Boolean formula evaluates to 1 for $(N-1)$ cases and to 0 in one case. Let's also assume that $|\psi_1\rangle$ is a state that indicates a tautology and $|\psi_2\rangle$ is the state that we derive. We need to unambiguously distinguish between these two non-orthogonal states.

Let's assume that M_0, M_1, and M_2 be three operators such that if we get the outcome M_1, we will be certain that the state was $|\psi_1\rangle$. If we get the outcome M_2, we will be certain that the state was $|\psi_2\rangle$. The outcome M_0 means that the result is inconclusive.

In other words, M_1 must never trigger for $|\psi_2\rangle$. M_2 must never trigger for $|\psi_1\rangle$. And, the following holds:

$$M_0 + M_1 + M_2 = I \tag{39}$$

Let $|\psi_1^{\perp}\rangle$ and $|\psi_2^{\perp}\rangle$ be the states orthogonal to $|\psi_1\rangle$ and $|\psi_2\rangle$, respectively. Therefore, we can construct M_1 and M_2 in the following way:

$$\begin{aligned} M_1 &= a|\psi_2^{\perp}\rangle\langle\psi_2^{\perp}| \\ M_2 &= b|\psi_1^{\perp}\rangle\langle\psi_1^{\perp}| \\ M_0 &= I - M_1 - M_2 \end{aligned} \tag{40}$$

We choose a and b in such a way that M_0 is minimized. We choose the following value for a and b:

$$a = b = \frac{1}{1 + |\langle\psi_1|\psi_2\rangle|} \tag{41}$$

Let's look at an example. Let's say the Boolean formula has two variables. It evaluates to 1 for 3 cases and to 0 in one case. The output state we obtain in this case is the following:

$$|\psi_2\rangle = \begin{bmatrix} 0.75 \\ 0.25 \\ 0.25 \\ -0.25 \\ 0.25 \\ -0.25 \\ -0.25 \\ 0.25 \end{bmatrix} \tag{42}$$

If the Boolean formula was a tautology, we would have obtained the following state:

$$|\psi_1\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{43}$$

Therefore, a and b, in our case, will be:

$$a = b = \frac{1}{1+\left|\langle\psi_1|\psi_2\rangle\right|} = \frac{1}{1.75} = 0.57 \tag{44}$$

Therefore, the probability of getting M_0 will be the following:

$$M_1 = a\left|\psi_2^{\perp}\right\rangle\left\langle\psi_2^{\perp}\right|$$

$$M_2 = b\left|\psi_1^{\perp}\right\rangle\left\langle\psi_1^{\perp}\right|$$

$$M_0 = I - M_1 - M_2$$

$$P(\text{failure}) = \langle\psi_1|M_0|\psi_1\rangle = 0.75 \tag{45}$$

Table 1 shows the probability of failures for different numbers of Boolean variables.

Table 1. Probability of failures.

Number of Boolean variables	Probability of failure	Minimum number of times the experiment should be repeated (k = a positive constant)
2	0.75	100
3	0.875	100
4	0.9375	100
5	0.96875	100
6	0.9843	$100 \times k$
7	0.9921	$100 \times k$
8	0.9960	$100 \times k$
9	0.9980	$100 \times k$
10	0.9990	$1000 \times k$
11	0.99951	$1000 \times k$
12	0.99975	$1000 \times k$
13	0.999877	$1000 \times k$

Therefore, we will get an inconclusive result 75% of the time, in which case we need to repeat the experiment. If the number of Boolean variables is 10, then the probability of getting an inconclusive result becomes 0.9990. In that case, if we repeat the experiment, for example, 1000 or 2000 times, in the worst case, we can correctly determine whether the Boolean formula is a tautology, satisfiable, or unsatisfiable.

Hence, if we repeat the experiment the required number of times, the proposed quantum algorithm can correctly predict whether the Boolean formula is a tautology, satisfiable, or unsatisfiable.

8. Time Complexity of The Proposed Quantum Algorithm

The value of $M_{2N\times 2N}$ is constant for a given value of $N=2^n$ where n is the number of Boolean variables in $\mathcal{F}$. Therefore, once M is constructed for a specific N, we can use the same value of M for all Boolean functions with n variables.

The proposed quantum algorithm must be run a constant number of times to test whether the corresponding qubits are consistently $|00\cdots 0\rangle$ or $|00\cdots 1\rangle$. If no, then the Boolean formula is satisfiable. If yes, then we need to repeat this experiment as described in Section 7.

Therefore, once we construct M, we can correctly determine whether the given Boolean function is a tautology, satisfiable, or non-satisfiable. And, construction M takes $O(n)$ time where n is the number of Boolean variables in the given Boolean function.

9. Conclusion

This paper proposes a quantum algorithm to solve the tautology and the satisfiability problems. In a classical computer, we may need to evaluate all 2^n assignments on n Boolean variables in the worst case. As a result, these problems can take $O(2^n)$ time for n Boolean variables. However, the proposed quantum algorithm shows significant improvement, as the stated problem can be solved in $O(1)$ time, in most cases, once M is constructed. Only when we get the result $|000\cdots 0\rangle$ or $|000\cdots 1\rangle$, we need to repeat the experiment as mentioned in the paper. Constructing M takes $O(n)$ time, where n is the number of Boolean variables of the given Boolean function.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Goldreich, O. (2010) P, NP, and NP-Completeness: The Basics of Computational Complexity. Cambridge University Press. https://doi.org/10.1017/CBO9780511761355

[2] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.

[3] Horn, R.A. and Johnson, C.R. (2012) Matrix Analysis. Cambridge University Press. https://doi.org/10.1017/CBO9781139020411

[4] Paris, M. and Rehacek, J. (2004) Quantum State Estimation. Springer Science & Business Media. https://doi.org/10.1007/b98673

[5] (2025) POVM. https://en.wikipedia.org/wiki/POVM