Application of Grover's quantum algorithm for string matching

Author: Júlia Barberà Rodríguez
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.

Advisor: Alba Cervera Lierta (Barcelona Supercomputing Center)
Tutor: Bruno Juliá Díaz (Universitat de Barcelona)

Abstract: In this work we present a quantum algorithm for exact string matching that relies on Grover's algorithm. Grover's algorithm, commonly used for unsorted data search, can be adapted to solve the problem and find a pattern's location within a string. This work contains the demonstration of Grover's algorithm for one and multiple target. It also presents the principles of quantum string matching, how to tackle this type of problem using Grover's algorithm and the detailed steps to construct the query. The quantum string matching algorithm is then implemented in Qibo, an open-source full stack API for quantum simulation and quantum hardware control. We explicitly expose an example for a string of length $N = 8$ and a pattern of length $M = 2$.

I. INTRODUCTION

String search in databases is a widely used resource these days and can be applied to many fields such as bioinformatics and DNA sequencing, spelling checking, plagiarism detection among others. It consists in finding the location of a pattern of length M within a longer string of length N such that $M \leq N$. Usually, the string length is very large and the pattern is not frequent in the text, so it involves large time complexity to find the position where the match occurs.

Kunth-Morris-Pratt and Boyer Moore algorithms [1] are the most common classical algorithms used for exact string matching. They check the characters from left to right until there is a match, thus, they will require at worst a processing time of $O(N + M)$. In this new era, where the quantum computation paradigm is in the ascendant, many problems that until now had been addressed with classical algorithms are being solved using quantum algorithms in order to reduce the number of queries. With the concern of improving the running time, we shall here explore the possibility to tackle the string matching problem using a quantum computer that takes advantage of quantum mechanics laws such as superposition, entanglement, and interference, to perform calculations.

String matching problem can be reformulated as a problem to search for the solution (the position in the string that matches the target) in a general database formed by all string positions.The best-known quantum algorithm for unsorted data search was proposed by Lov K. Grover in 1996 and offers a quadratic speed-up in query complexity $O(\sqrt{N})$ [2]. The strength of this algorithm lies within Grover's oracle which is able to find if the solution is in the given database by employing reflection (a phase flip of $e^{i\pi}$) to mark the target. The diffusion operator, which is the second component of Grover's algorithm, is based on amplitude amplification, a method that provides to measure the solution with the highest probability among all the other elements' probabilities.

The motivation of this work is to demonstrate and extend the method used by P.Mateus and Y.Omar in [3] for quantum pattern matching. With the purpose of understanding the principles of Grover's algorithm in-depth, we will construct the query function for this type of problem and the diffusion operator. The number of qubits needed to solve the string matching problem escalates as $N + M + s \cdot M$, with $s = \lceil \log_2 N - M \rceil$ to define the qubits needed to represent each index.

The work is structured as follows. Section II is devoted to explaining Grover's algorithm for one and multiple targets and the optimal number of iterations needed to find the target state with the highest probability. In section III, we introduce the quantum string matching algorithm, the detailed steps to construct the oracle and the diffusion operator. In the next section, we implement the full quantum algorithm, using quantum logic gates and in section IV we show and discuss the results obtained for general quantum string search, using $N = 8$ and $M = 2$. Finally, we present the main conclusions based on the results obtained.

II. GROVER'S ALGORITHM

In this section, we present a general version of Grover's algorithm for finding one and multiple target states on an unsorted database following Ref. [4].

Let us consider a Hilbert subspace of dimension $N = 2^n$, where n is the number of qubits. From the possible 2^n states, assume $|w\rangle$ is the one that needs to be found. The algorithm will act as follows. It starts with the uniform superposition of all possible basis states:

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle, \tag{1}$$

where $|w\rangle$ will be one of them. We assume that these computational basis states are orthogonal, implying that $\langle w'|w\rangle = 0$ if $w' \neq w$ and $\langle w'|w\rangle = 1$ if $w' = w$. In the beginning we don't know that $|w\rangle$ is the solution so the probabilities to measure any state of this basis will be equal for all of them.

Grover's relies on an oracle, a unitary operation that marks those states that represent the solution of the problem. This oracle is applied after the superposition and performs a reflection of $|w\rangle$ as:

$$O = I - 2|w\rangle\langle w|, \tag{2}$$

which acts like:

$$\begin{cases} O|x\rangle = -|x\rangle \;\; for \;\; x = w \\ O|x\rangle = |x\rangle \;\; for \;\; x \neq w \end{cases}$$

Notice that, the states still have the same probability, we only have performed a reflection. The state after applying the oracle becomes:

$$|\psi\rangle = O|\psi\rangle = (I - 2|w\rangle\langle w|)\,|\psi\rangle = |\psi\rangle - \frac{2}{\sqrt{N}}|w\rangle. \tag{3}$$

This oracle is specific for every computational problem and its explicit form requires to be found for each case.

After the reflection operator we apply the diffusion operator, also named the amplification operator. It performs a reflection about the average amplitude,

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I. \tag{4}$$

Applying it to the resulting state from Eq.(3), it becomes

$$(2|s\rangle\langle s| - I)\,|s'\rangle = \frac{1}{\sqrt{N^3}}\left((N-4)\sum_{x\neq w}|x\rangle + (3N-4)|w\rangle\right). \tag{5}$$

The coefficient of $|w\rangle$ is greater than the other states from the basis, which means that the probability of measuring $|w\rangle$ will be greater than the others. In particular, the probability becomes

$$P(|w\rangle) = \left(\frac{3N-4}{N\sqrt{N}}\right)^2. \tag{6}$$

This algorithm can be extrapolated to search for more than one element of a list. Considering that the number of solutions is M, the space, for this case, would be spanned by the following orthogonal vectors:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_{x''}|x\rangle \quad \text{and} \quad |\beta\rangle = \frac{1}{\sqrt{M}}\sum_{x'}|x\rangle,$$

where x' are all the states x that are solution of the problem, and x'' the rest.

Proceeding in the same way as before, the target elements would be found with a probability of

$$P(|w\rangle) = \frac{1}{N}\left(3 - \frac{4M}{N}\right)^2. \tag{7}$$

The optimal number of iterations to measure the target state with the highest probability can be computed by considering the two orthogonal vectors used before: $|\alpha\rangle$ and $|\beta\rangle$. The initial state $|\psi\rangle$ can also be expressed as:

$$|\psi_0\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle, \tag{8}$$

with $\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}}$ and $\sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$. If we apply Grover's iterator, $G = (2|\psi\rangle\langle\psi| - I)O$, k times to $|\psi\rangle$, the wave function results into the following superposition:

$$G^k|\psi_0\rangle = \cos\frac{2k+1}{2}\theta|\alpha\rangle + \sin\frac{2k+1}{2}\theta|\beta\rangle. \tag{9}$$

We denote the optimal number of iterations as:

$$R = CI\left(\frac{\arcsin\sqrt{M/N}}{\theta}\right), \tag{10}$$

where CI stands for the closest integer to its argument. If $M \ll N$, $\theta \approx \sin\theta \approx 2\sqrt{\frac{M}{N}}$. The maximum amplitude of the target is obtained when the rotation angle is: $\theta_{max} = \frac{\pi}{2}$, hence, $(2k+1)\sqrt{\frac{M}{N}} \leq \frac{\pi}{2}$. Considering the Taylor series of $\arcsin x \simeq x + O(x^3)$ when $N \gg M$, the relation found is

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil. \tag{11}$$

III. QUANTUM STRING MATCHING

The problem of finding the occurrence of a pattern within another string can be solved using Grover's algorithm, as it is presented in [3].

Let us assume that we want to find where the pattern $|p\rangle$ of size M occurs within the string $|w\rangle$ of size N. To return the position $|i\rangle \in N$ we have to encode it in a unit vector of a Hilbert subspace of dimension N: $|i\rangle = \{|1\rangle, ..., |N\rangle\}$. The second match will have to occur just after the first one and so on as $|i\rangle\otimes|i+1\rangle\otimes...|N-M+1\rangle$. Consequently, the initial state for the indexes set consists of a uniform superposition of all the possible solutions expressed as

$$|\psi_0\rangle = \frac{1}{\sqrt{N-M+1}}\sum_{i=0}^{N-M+1}(|i\rangle\otimes|i+1\rangle\otimes...\otimes|i+M-1\rangle). \tag{12}$$

We will use a decimal number to represent the index location, which at the same time, can be expressed in binary. Each zero and one of these binary numbers will be a qubit in that state. For instance, the state $|i\rangle = |7\rangle$ represents the position 7 within the string $|w\rangle$ which in binary becomes $|111\rangle$. The minimum number of bits needed to represent each index is given by $s = \lceil \log_2 N - M \rceil$. If the pattern had more characters we should add s qubits for each extra digit of the pattern. Thus, the total circuit qubits will escalate as $N + M + s \cdot M$.

A. The Oracle

Then, we need to implement Grover's oracle. From now on, the oracle will be expressed as U_σ to be consistent with expressions in [3]. Thus, σ represents every pattern's symbol and we can express the set of σ's as Σ, which is named the alphabet of the problem. In our case, the pattern will be in binary so $\sigma = 0$ or $\sigma = 1$ and $\Sigma = \{0, 1\}$.

The oracle will mark the state when a match occurs. Hence, the oracle will take the first string character and identify if there is a match with the pattern's symbol. If they are equal, the state corresponding to the first index will be flipped, otherwise, the oracle will leave the state unchanged. Then, the following string character will be compared with the symbol and the second index state will be flipped if they match. At the end of this process, the initial state from Eq.(12) will have some states marked corresponding to the locations where a string character has matched the pattern's symbol. More formally, we can express this oracle as

$$U_\sigma(|i\rangle \otimes |i+1\rangle) = (-1)^{f_\sigma(i)}(|i\rangle \otimes |i+1\rangle), \tag{13}$$

with, $f_\sigma(i) = 1$ if the i-th character of $|w\rangle$ is σ and $f_\sigma(i) = 0$ otherwise. The oracle for each pattern's symbol needs to be applied as many times as the symbol occurs in the pattern as

$$|\psi\rangle = \prod_{\sigma \in |p\rangle} U_\sigma |\psi_0\rangle. \tag{14}$$

For example, if $|p\rangle = |0110\rangle$ then, $|\psi\rangle = U_0U_1U_1U_0|\psi_0\rangle$.

The next step is to amplify the states that are solution using the diffusion operator as shown in Eq.(4), where $|\psi\rangle = \sum_{i=0}^{N-M+1} \frac{1}{\sqrt{N}}|i\rangle$ and I is the identity operator of dimension $N - M + 1$.

B. Example, string with $N = 8$ and pattern $M = 2$

Let us assume that we are given a string $|w\rangle = |11100000\rangle$ of length $N = 8$ and we need to find the location of a pattern $|p\rangle = |10\rangle$ of length M.

The initial state for this case will be

$$|\psi_0\rangle = \frac{1}{2\sqrt{2}}(|0,1\rangle + |1,2\rangle + |2,3\rangle + |3,4\rangle + |4,5\rangle + \\ + |5,6\rangle + |6,7\rangle + |7,0\rangle). \tag{15}$$

To simplify the notation we have written the position states $|i\rangle \otimes |j\rangle$ as $|i,j\rangle$. Thus, the state $|0,1\rangle$ stands for the pattern's sub-string 11, $|2,3\rangle$ for 10 and so on. Notice that we have used Periodic Boundary Conditions for convenience to use Hadamard gates to generate the initial state.

As our pattern is $|p\rangle = |10\rangle$, we will need to apply U_1 and U_0 once. Starting with the first symbol,

$$U_1|\psi_0\rangle = \frac{1}{2\sqrt{2}}(-|0,1\rangle - |1,2\rangle - |2,3\rangle + |3,4\rangle + |4,5\rangle + \\ + |5,6\rangle + |6,7\rangle + |7,0\rangle), \tag{16}$$

since the $|1\rangle$ state appears in position $0, 1$ and 2. Then, after applying the oracle for the second symbol:

$$|\psi\rangle = U_0U_1|\psi_0\rangle = \frac{1}{2\sqrt{2}}(-|0,1\rangle - |1,2\rangle + |2,3\rangle - |3,4\rangle - \\ - |4,5\rangle - |5,6\rangle - |6,7\rangle + |7,0\rangle). \tag{17}$$

States in Eq.(17) that are positive should be the problem's solutions. Although the state $|2,3\rangle$ has a positive sign and corresponds to the target, the last state is not the solution and it is also positive. This state has been marked with neither of the oracles and when D_N is applied we will measure this state with the same probability as the solution. This problem can be solved by applying the diffusion operator (Eq.(4)) right after every oracle. After implementing the first oracle, the diffusion operator will amplify the first three states. Then, when the U_0 is applied, the target state will be more amplified than the others, since its amplitude was already greater.

IV. IMPLEMENTATION

In this section, we show the explicit implementation of the algorithm. We have programmed this example in Qibo [5], an open-source full stack API for quantum simulation and quantum hardware control.

A. Circuit initialization

Initially, all the qubits will be set to $|0\rangle$ except the string and pattern symbols that are in the state 1 which will be set to $|1\rangle$. To create the initial state from Eq. (12) we need to proceed as follows. Firstly we create the index state $\sum_{i=0}^{N-M+1} |i\rangle \otimes |i\rangle$. To do so, we apply Hadamard gates to the first set of index qubits. A Hadamard gate is expressed with the following matrix,

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{18}$$

and is used to create a superposition of all the states with equal amplitude. Once the H gate is applied, the state obtained is

$$|\psi\rangle = \frac{1}{\sqrt{N-M+1}} \sum_{i=0}^{N-M+1} (|i\rangle + ... + |i+M-1\rangle) \otimes |0\rangle^{\otimes s\cdot(M-1)} \tag{19}$$

Then, we need to entangle the first set of qubits, s_1, that represents the position of the first symbol with the

remaining sets of $s = s_2, s_3, ...s_M$ that represent the other indexes using CX (CNOT) gate, which is given by

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{20}$$

These gates rotate the target qubit if the control qubit is at state $|1\rangle$. The s_1 qubits will act as control qubits and the second set as targets as it is represented in Fig. 1. The resultant state is $\sum_{i=0}^{N-M+1} |i\rangle \otimes |i\rangle$.
Finally, we need to increment by 1 the second state which can be achieved by implementing a series of MCX gates over more than one qubit (as demonstrated in [6]), rotating the target if the controls are in state $|1\rangle$.

An example of the initialization circuit for $N = 8$ and $M = 2$ is shown in Fig.1.

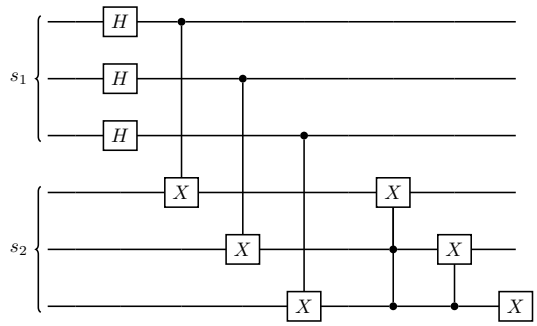

FIG. 1: Initial circuit to create state $|\psi_0\rangle$ for $N = 8$ and $M = 2$ given by Eq.(12), using quantum gates. The first set s_1 represents the first string's position and the second set s_2 encodes the second position. If the pattern had more symbols, we should add as many s sets to the circuit as extra pattern symbols.

B. Grover's oracle: U_σ

The oracles will be built up using X gates and MCZ gates which are represented by the following matrices:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad MCZ = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix}. \tag{21}$$

If the pattern's symbol is in $|0\rangle$, X gates will be applied before and after the control to perform the transformation if the qubit is in $|0\rangle$. For every symbol of the pattern, the oracle acts as follows:

1. Checks if the symbol matches every string character using controls.

2. Applies X gates to every set of s qubits if their qubits are in $|0\rangle$, bringing them to the last entry of a MCZ matrix.

3. MCZ gate is implemented to flip the target qubit using as controls the character qubit, the symbol qubit, and the first two qubits of s. The last qubit of the set s is the target.

4. If all controls are in the same state, the index state is marked ($e^{i\pi}$ phase).

If we consider the first symbol, the first set of s qubits will be used for the process. To check the second symbol, we need to use the second set of s qubits and so on.

Considering an arbitrary string of length $N = 6$ and a pattern $|p\rangle = |10\rangle$ ($M = 2$), we present an example in Fig.2 where we compare if the first symbol of the string matches the first pattern's symbol that is in the state $|1\rangle$ and if the second string's symbol is in the state $|0\rangle$ as the second pattern's symbol.

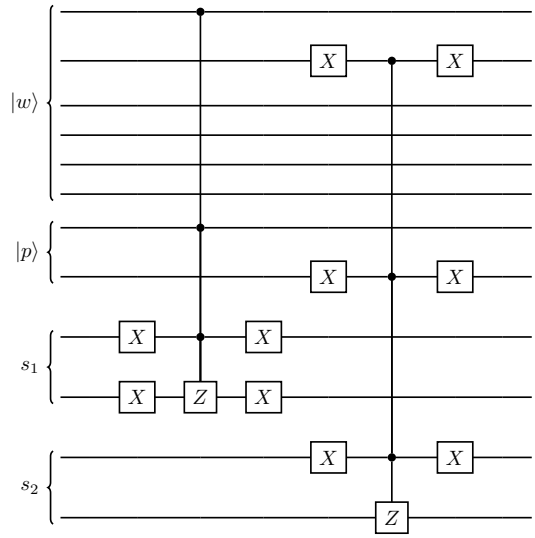

FIG. 2: Part of the oracle's circuit that compares the first two symbols of the string $|w\rangle$ with the two symbols of the pattern $|p\rangle$, using quantum gates with $N = 6$ and $|p\rangle = |10\rangle$ ($M = 2$). If the symbol of the pattern is in the state $|0\rangle$, we have to apply X gates before and after the target to do the comparison.

C. Diffusion operator: D_N

Once the oracle for the first symbol is applied, the algorithm applies the diffusion operator to achieve a highest amplitude for the future solution. This operator, represented by Eq.(4), can be expressed using H, X, and MCZ gates as:

$$D_N = H^{\otimes s \cdot M} \cdot X^{\otimes s \cdot M} \cdot MCZ \cdot X^{\otimes s \cdot M} \cdot H^{\otimes s \cdot M}. \tag{22}$$

The H gates bring $|\psi\rangle$ to $|000\rangle$. Then, $X^{\otimes s \cdot M} \cdot MCZ \cdot X^{\otimes s \cdot M}$ performs the reflection and the H gates at the end place the states to their original state.

V. RESULTS AND DISCUSSION

We show here the examples using two different strings: $|w_1\rangle = |11100000\rangle$ and $|w_2\rangle = |10001000\rangle$, and one pattern: $|p\rangle = |10\rangle$. Thus, $N = 8$, $M = 2$ and the total number of qubits used is 16. Using $nshots = 10000$ we obtain the probabilities shown in Fig.3.

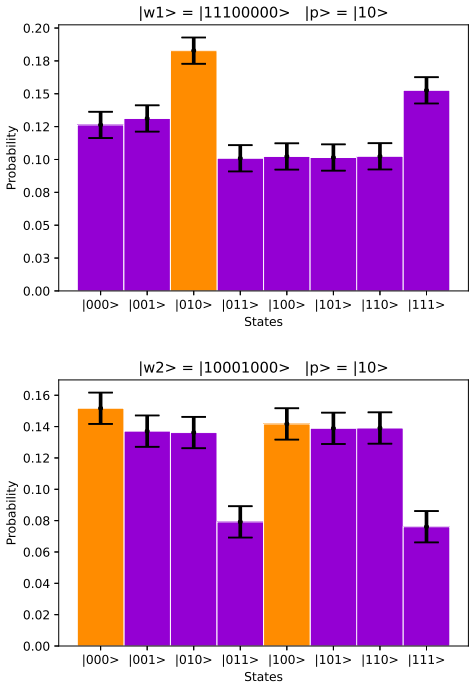

FIG. 3: Results obtained after implementing the quantum algorithm using $N = 8$ and $M = 2$, for $|w_1\rangle$ (upper panel) and $|w_2\rangle$ (lower panel). States in orange represent the solutions, found with the highest amplitude. The error bars are given by $1/\sqrt{nshots}$.

For the first case, only one Grover's iteration has been required to find the solution with a maximum probability of 0.19. Although the solution can be distinguished from the other states, this value is not high enough and we will only obtain the target 20% of the times we do the measurement. For the second case, the pattern has been found twice along the string using two iterations. Nevertheless, the amplitudes are almost indistinguishable. The inaccuracy of these results could be associated to the fact that we are searching for one/two solutions of length $M = 2$ within a string of length $N = 8$, therefore, is not satisfied that $N \gg M$. To solve this problem we could use a larger string but this solution would involve more qubits.

VI. CONCLUSIONS

In this work, we have implemented a quantum search algorithm as proposed in [3] using Grover's algorithm and presented the detailed steps to construct our oracle. We have shown the explicit gate construction and programmed it using Qibo language for $N = 8$ and $M = 2$. While the classical algorithm takes at worst $N+M$ queries to find the solution, the index location has been found with only one/two queries using the quantum algorithm. This shows an improvement in the running time and the advantage of using a quantum algorithm over a classical one. However, the string and pattern length involve a limitation on the algorithm ($N \gg M$) as has been discussed in the previous section due to the inaccurate results.

Some proposals to continue this work could be to generalize this algorithm to the fourth dimension to search for codons, tripletes of the nitrogenous basis letters (A, C, G, U), in a DNA sequence. We could use ququarts, qudits of dimesion 4, to solve the problem. For this case, we should have to change the alphabet to $\Sigma = \{A, C, G, U\}$ with $A = |0\rangle, C = |1\rangle, G = |2\rangle, U = |3\rangle$ and while the diffusion operator would remain unchanged, we should adapt the oracle to operate on ququarts.

Code Availability

The tutorials for the Grover's algorithm and quantum search algorithm can be found at:

`Github: juliabarbera/TFG-quantum-string-matching`

Acknowledgments

I would like to thank Alba Cervera Lierta for the time and passion she has devoted to teach me quantum computing from scratch and for her enlightening guidance during this project and also to BSC for hosting me.

[1] Mandumula, K. K., *Knuth-Morris-Pratt.* Indiana State University Terre Haute IN (2011).
[2] Grover, L. K., *A fast quantum mechanical algorithm for database search*, arXiv: quant-ph/9605043 (1996).
[3] Mateus, P. and Omar, Y., *Quantum pattern matching.*,arXiv: quant-ph/0508237 (2005).
[4] Nielsen, M.A. and Chuang, I.L., *Quantum computation and quantum information.*, Phys. Today 54.2 (2001): 60.
[5] Efthymiou, S., et al., *Qibo: a framework for quantum simulation with hardware acceleration.*, Quantum Science and Technology 7.1 (2021): 015018.
[6] Li, X., et al., *A class of efficient quantum incrementer gates for quantum circuit synthesis.*, International Journal of Modern Physics B 28.01 (2014): 1350191.