Practical Quantum Computing

Lecture 8
Quantum Algorithms: Simon's and Grover

with slides from Dave Bacon https://homes.cs.washington.edu/~dabacon/teaching/siena/

Context

The Deutsch-Jozsa problem showed an exponential quantum improvement over the *best deterministic* classical algorithms.

The Bernstein-Vazirani problem shows a polynomial improvement over the *best randomized* classical algorithms that have error probability ≤ 1/3.

Combine these two features and see a problem where quantum computers are exponentially more efficient than bounded-error randomized algorithms.

https://arxiv.org/pdf/1907.09415.pdf

Simon's Problem

Given: A function with n bit strings as input and one bit as output

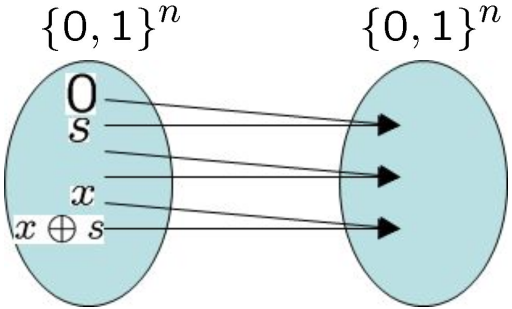
$$f : x \in \{0,1\}^n \rightarrow \{0,1\}^n$$

Promise: The function is guaranteed to satisfy

$$f(x) = f(y) \Leftrightarrow y = x \oplus s, s \neq 0$$

$$x \oplus s = (x_1 \oplus s_1, x_2 \oplus s_2, \ldots, x_n \oplus s_n)$$

Problem: Find the n bit string $s \neq 0$



Classical Simon's Problem

Promise: The function is guaranteed to satisfy

Suppose we start querying the function and *build up a list of the pairs* $(x_\alpha, f(x_\alpha))$

If we find $x_\alpha \neq x_\beta$ such that $f(x_\alpha) = f(x_\beta)$ then we solve the problem

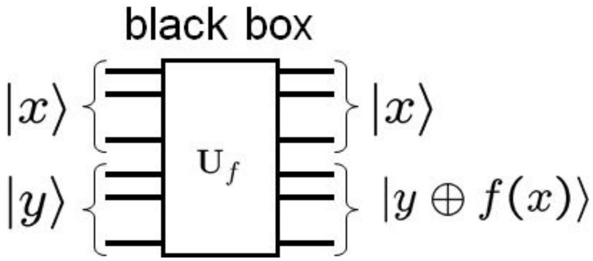
$$f(x_\alpha) = f(x_\beta) \Rightarrow x_\alpha = s \oplus x_\beta$$
$$s = x_\alpha \oplus x_\beta$$

But suppose we start querying the function m times

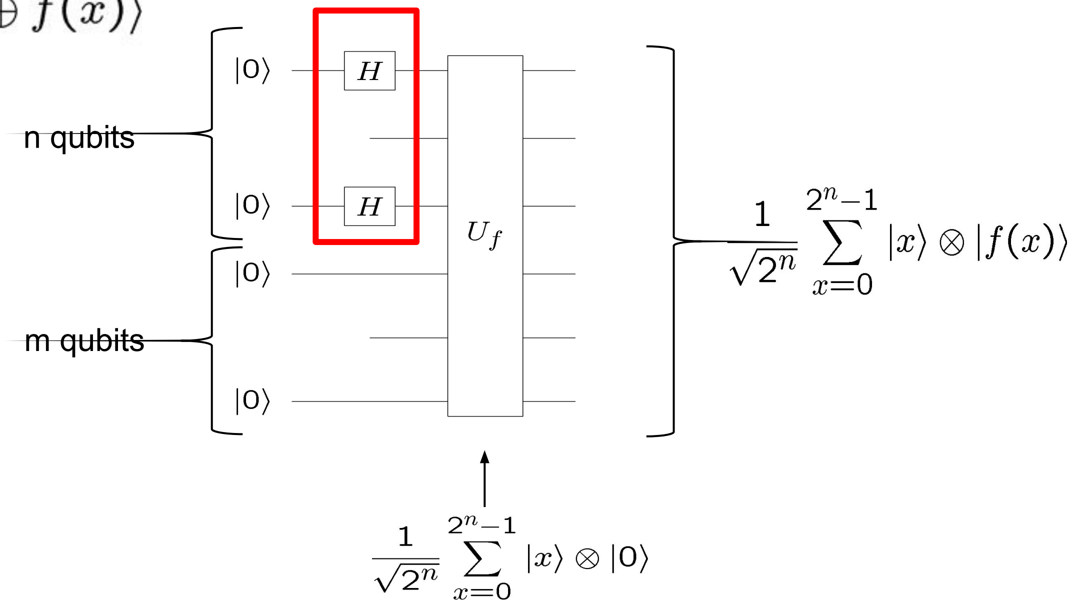
Probability of getting a matching pair: $\approx \frac{\binom{m}{2}}{2^n} = \frac{m(m-1)}{2^{n+1}}$

Bounded error query complexity: $m = O\left(2^{\frac{n}{2}}\right)$

Quantum Simon's Problem



Unlike previous problems, we can't use the phase kickback trick because there is no structure in the function.



Quantum Simon's Problem

Measure the second register

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

Using the promise on the function

$$f(x) = f(y) \Leftrightarrow y = x \oplus s, s \neq 0$$

This implies that after we measure, we have the state

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle) \otimes |f(x)\rangle$$

For random uniformly distributed $x \in \{0,1\}^n$

uniformly distributed = all strings equally probable.

measuring this state at this time does us no good ...

Quantum Simon's Problem

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$$

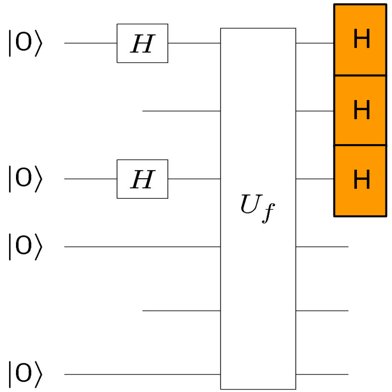
Measuring this state at this time in the **computational basis** does us no good….

For random uniformly distributed $x \in \{0,1\}^n$

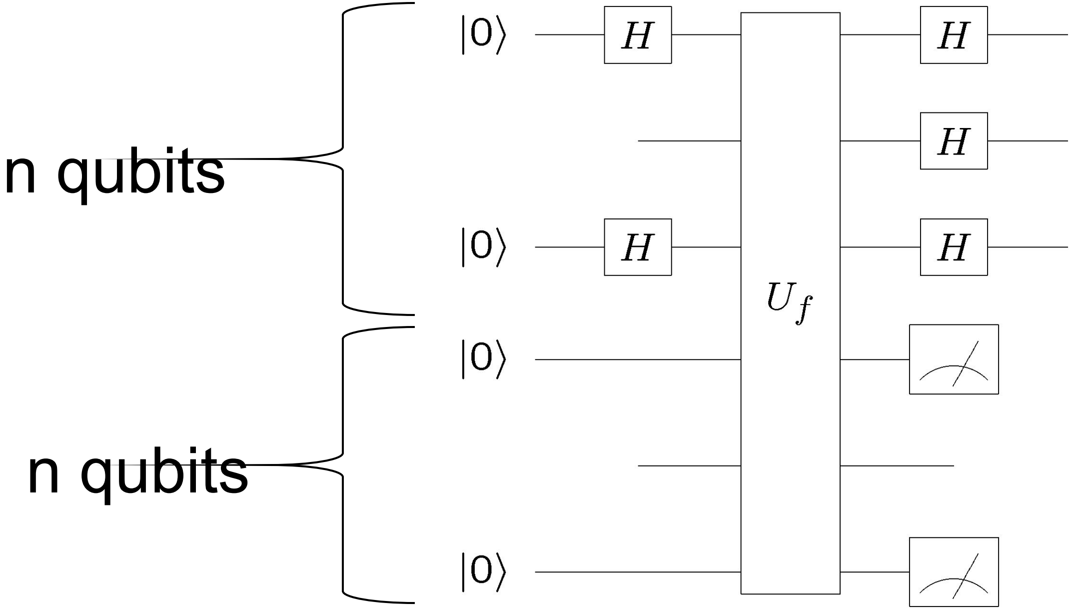
Measurement yields either $|x\rangle$ or $|x \oplus s\rangle$

But we don't know x, so we can't use this to find s.

Add Hadamard gates to the end register



Quantum Simon's Problem



$$H^{\otimes n}\frac{1}{\sqrt{2}}(|x\rangle+|x\oplus s\rangle) = \frac{1}{\sqrt{2^{n+1}}}\sum_{y=0}^{2^n-1}((-1)^{y\cdot x}+(-1)^{y\cdot(x\oplus s)})|y\rangle$$

Quantum Simon's Problem

$$H^{\otimes n}\frac{1}{\sqrt{2}}(|x\rangle+|x\oplus s\rangle) = \frac{1}{\sqrt{2^{n+1}}}\sum_{y=0}^{2^n-1}((-1)^{y\cdot x}+(-1)^{y\cdot(x\oplus s)})|y\rangle$$

$$y\cdot x = y_1x_1 \oplus y_2x_2 \oplus \cdots \oplus y_nx_n$$

$$y\cdot(x\oplus s) = y_1(x_1\oplus s_1)\oplus y_2(x_2\oplus s_2)\oplus\cdots\oplus y_n(x_n\oplus s_n) = (y\cdot x)\oplus(y\cdot s)$$

$$= \frac{1}{\sqrt{2^{n+1}}}\sum_{y=0}^{2^n-1}(-1)^{y\cdot x}\boxed{(1+(-1)^{y\cdot s})}|y\rangle$$

Measuring this state, we obtain uniformly distributed random values of y s.t.

$$y\cdot s = 0 \text{ mod } 2$$

If $y \neq 0$ we have eliminated the possible values of s by half

Quantum Simon's Problem

$$y \cdot s = 0 \bmod 2$$

$$y \cdot s = y_1 s_1 \oplus y_2 s_2 \oplus \cdots \oplus y_n s_n = 0$$

On values of y_i which are 0, this doesn't restrict s_i

On values of y_i which are 1, the corresponding s_i must XOR to 0.

This restricts the set of possible $\boldsymbol{s}$'s by half.

Example: $n = 3$

$$y = 3_{10} = 011_2$$

$$(011) \cdot s = 0s_1 \oplus 1s_2 \oplus 1s_3 = 0$$

$$s_2 \oplus s_3 = 0$$

possible s's: 000 011 100 111

Quantum Simon's Problem

Think about the bit strings s as vectors in $\mathbb{Z}_2^n$

$$y_0 \cdot s = 0$$
$$y_1 \cdot s = 0$$
$$\vdots$$
$$y_k \cdot s = 0$$

- If we obtain **n** lin. indep. equations of this form, we win
- (Gaussian elimination)

Suppose we have k linearly independent y_i 's. What is the probability that y_{k+1} is linearly independent of previous y_i 's?

$$Pr = \frac{2^n - 2^k}{2^n} = 1 - 2^{k-n}$$

Note that if the j's you have generated at some point span a space of size 2^k, for some $k < n-1$, then the probability that your next run of the algorithm produces a j that is linearly independent of the earlier ones, is $(2^{n-1} - 2^k)/2^{n-1} \geq 1/2$. Hence an expected number of $O(n)$ runs of the algorithm suffices to find $n-1$ linearly independent j's. Simon's algorithm thus finds s using an expected number of $O(n)$ x_i-queries and polynomially many other operations.

https://arxiv.org/pdf/1907.09415.pdf

Quantum Simon's Problem

What is the probability that our n-1 equations are linearly independent?

$$Pr(succ) = \left(1 - \frac{1}{2^n}\right)\left(1 - \frac{1}{2^{n-1}}\right)\cdots\left(1 - \frac{1}{4}\right)$$

$$= \prod_{k=1}^{n}\left(1 - \frac{1}{2^k}\right) > \prod_{k=2}^{\infty}\left(1 - \frac{1}{2^k}\right) \approx 0.28879$$

With constant probability:

- we obtain linearly independence -> Gaussian elimination O(n^3)
- solve Simon's problem

Applications of Grover's Algorithm

Grover's algorithm is a framework

- It does not offer the exponential speedup like Shor's alg.
- Can be extended for different problems
  - cryptanalysis AES
  - combinatorial optimisation - e.g. travelling salesman

Applying Grover's algorithm to AES: quantum resource estimates

Markus Grassl[1], Brandon Langenberg[2], Martin Roetteler[3], and Rainer Steinwandt[2]

[1] Universität Erlangen-Nürnberg & Max Planck Institute for the Science of Light, Günther-Scharowsky-Straße 1, Bau 24, 91058 Erlangen, Germany, Markus.Grassl@fau.de
[2] Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, U.S.A., {blangenb, rsteinwa}@fau.edu
[3] Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A., martinro@microsoft.com

Abstract. We present quantum circuits to implement an exhaustive key search for the Advanced Encryption Standard (AES) and analyze the quantum resources required to carry out such an attack. We consider the overall circuit size, the number of qubits, and the circuit depth as measures for the cost of the presented quantum algorithms. Throughout, we focus on Clifford+T gates as the underlying fault-tolerant logical quantum gate set. In particular, for all three variants of AES (key size 128, 192, and 256 bit) that are standardized in FIPS-PUB 197, we establish precise bounds for the number of qubits and the number of elementary logical quantum gates that are needed to implement Grover's quantum algorithm to extract the key from a small number of AES plaintext-ciphertext pairs.

Keywords: quantum cryptanalysis, quantum circuits, Grover's algorithm, Advanced Encryption Standard

Implementing **Grover** oracles for quantum key search on **AES** and LowMC
S Jaques, M Naehrig, M Roetteler, F Virdia - ... International Conference on ..., 2020 - Springer
... Keywords. Quantum cryptanalysis **Grover's** algorithm **AES** LowMC Post-quantum cryptography Q# implementation ... Since the publication of [21], other works have studied quantum circuits for **AES**, the **AES Grover** oracle and its use in **Grover's** algorithm. Almazrooie et al ...
☆ 99 Cited by 31 Related articles All 9 versions

[HTML] **Grover** on $SIMON$ SIMON
R Anand, A Maitra, S Mukhopadhyay - Quantum Information Processing, 2020 - Springer
... However, this does not rule out the need of analyzing the cost of **Grover's** algorithm on symmetric ciphers. In this direction, subsequent efforts have been made to derive cost estimation for applying **Grover's** search algorithm on all variants of **AES** [7, 11, 17, 28] ...
☆ 99 Cited by 5 Related articles All 6 versions

[PDF] **Grover** on SPECK: quantum resource estimates
K Jang, S Choi, H Kwon, H Seo - eprint.iacr.org
... computing, pp. 212–219, 1996. 6. M. Grassl, B. Langenberg, M. Roetteler, and R. **Steinwandt**, "Applying **Grover's** algorithm to **AES**: quantum resource estimates," in Post-Quantum Cryptography, pp. 29–43, Springer, 2016. 7. B ...
☆ 99 Cited by 4 Related articles

[PDF] Observations on the Quantum Circuit of the SBox of **AES**.
J Zou, Y Liu, C Dong, W Wu, L Dong - IACR Cryptol. ePrint Arch., 2019 - eprint.iacr.org
... [3] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer **Stein- wandt** ... TimeCspace complexity of quantum search algorithms in symmetric cryptanalysis: applying to **AES** and SHA-2. Quantum Information ... 8] Brandon Langenberg, Hai Pham, and Rainer **Steinwandt** ...
☆ 99 Cited by 2 Related articles

Quantum Resource Estimates of **Grover's** Key Search on ARIA
AK Chauhan, SK Sanadhya - International Conference on Security, Privacy ..., 2020 - Springer
... [10] studied the quantum circuits of **AES** and estimated the cost of quantum resources needed to apply **Grover's** algorithm to the **AES** oracle for key search. Almazrooie et al ... As a working example, they implemented the **AES Grover** oracle in Q# quantum programming language ...
☆ 99 Related articles

Solving Binary $\mathcal{MQ}$ with **Grover's** Algorithm
P Schwabe, B Westerbaan - ... Conference on Security, Privacy, and Applied ..., 2016 - Springer
... primitives. For example, in [GLRS16], Grassl, Langenberg, Roetteler, and **Steinwandt** describe how to attack **AES**-128 with **Grover's** algorithm using a quantum computer with 2953 logical qubits in time about \(2^{87}\). We note ...
☆ 99 Cited by 25 Related articles All 12 versions

Quantum **Grover** Attack on the Simplified-**AES**
M Almazrooie, R Abdullah, A Samsudin... - Proceedings of the 2018 ..., 2018 - dl.acm.org
... This paper is organized as follows: Sections 2 and 3 review the Simplified-**AES** (S-**AES**) cryptosystem and the quantum **Grover's** algorithm, respectively ... Figure 8. Applying **Grover** attack on S-**AES**. Figure 8 illustrates the complete model of the **Grover** attack against S-**AES** ...
☆ 99 Related articles

Borbely E. Grover search algorithm. arXiv preprint arXiv:0705.4171. 2007 May 29. - step by step derivation of Grover iterations

Quantum computers can search faster than a classical ones

- Assume the entries are indexed 0, 1, 2, 3, …., N

- Use binary vectors
  - Of the form 0 = 10….000, 1 = 01….000, … , N = 00...001
  - The length of the vectors is N bits
  - A bit signals if an entry is found in the database
  - Practically, multiple entries can be sought and then multiple bits will be on

$$\begin{array}{c} |0\rangle \\ |1\rangle \\ |2\rangle \\ |3\rangle \end{array} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- E.g. the vector |3> will have a 1 at the fourth index (zero-indexed)

- Search: “Is the entry with index F in the list 0,1,…..,N?”

- Simplify and assume that the search is always for F=N (relabel the database entries)

Building block - Inner product

The why: We will use this notation to determine the relative rotation (angle) between two vectors

Example: a = (0, 0, 0, 1) b = (1, 1, 1, 1) -> ab = 0*1 + 0*1 + 0*1 + 1*1 = 1

Can be written as the multiplication of a row vector with a column vector

$$(0, 0, 0, 1)\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0*1 + 0*1 + 0*1 + 1*1 = 1$$

Depending if the vector is row or column we can use special notation

<a| for row vector

|a> for column vector

such that <a||b> is the notation for the inner product

Shorthand notation <a|b> = 1

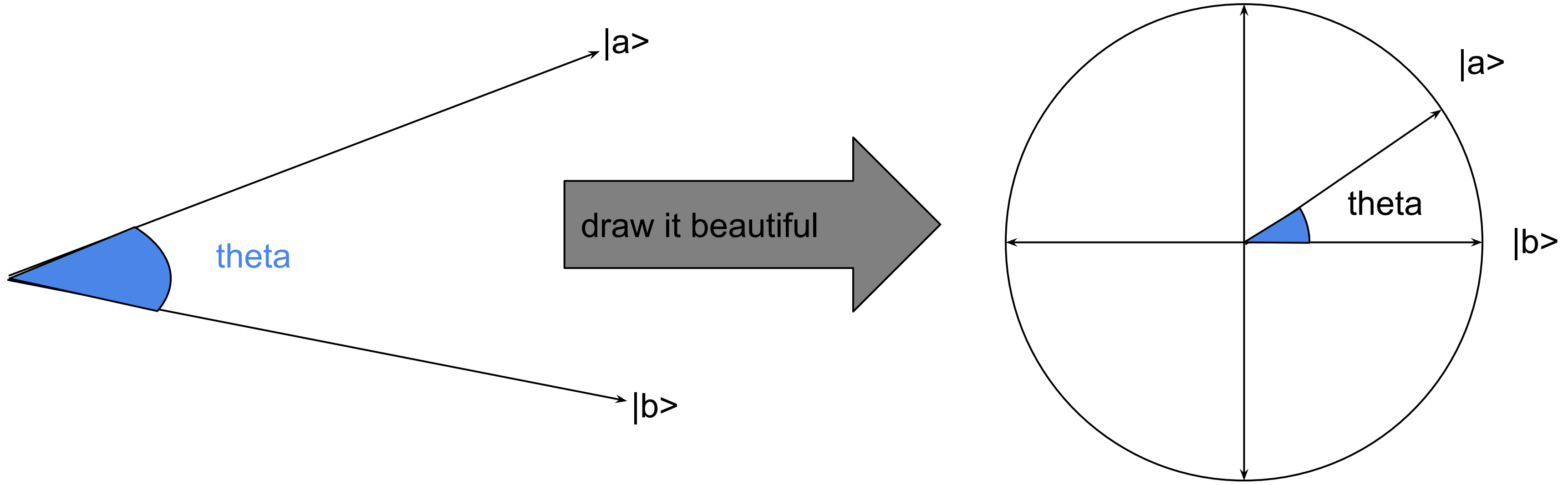
Building block - Angle between vectors

The why: Useful for building a method to rotate a third vector by knowing the angle between two other vectors

In general, <a|b> = |a||b|cos(theta)

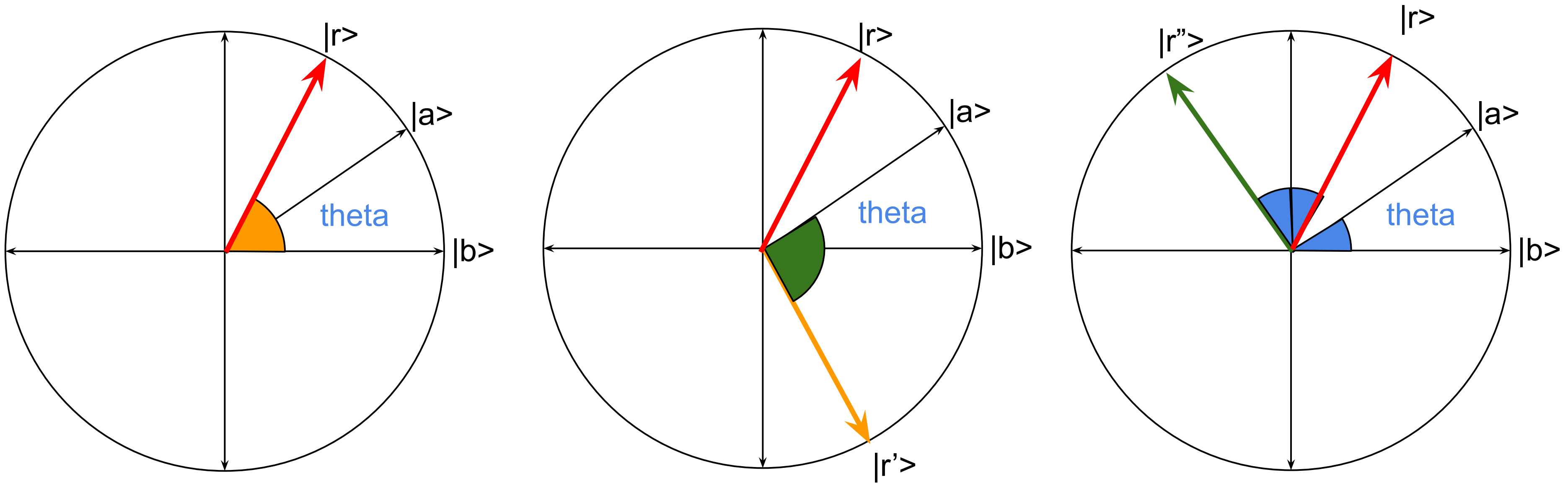
where |a| and |b| are the length of the vectors

Simplify and assume that all vectors have unit length, such that <a|b> = cos(theta)



Building block - rotate with twice the angle of theta

The why: Build a method to move |r> such that it is orthogonal to |b>



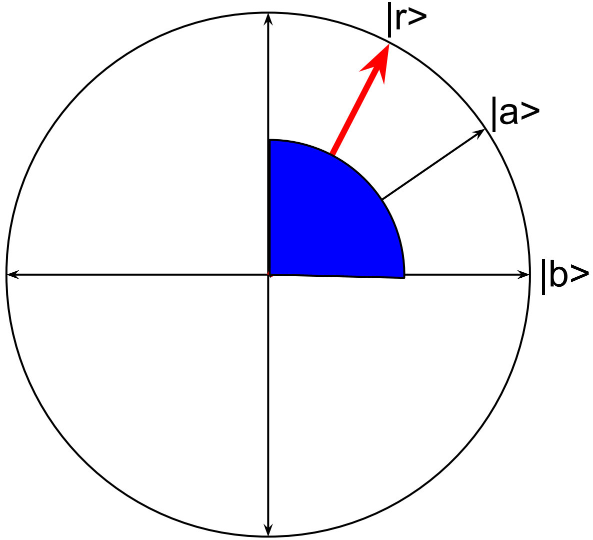
after mirror against |b>

after mirror against |a>

Building block - Number of mirror seq. to rotate pi/2

The why: It indicates where the quantum speed-up is coming from!

How many rotations (k=?) are necessary to get to pi/2?



theta + **k***2*theta = pi/2

2**k** = pi/(2 * theta) - 1

k = round(pi/(4*theta) - ½)

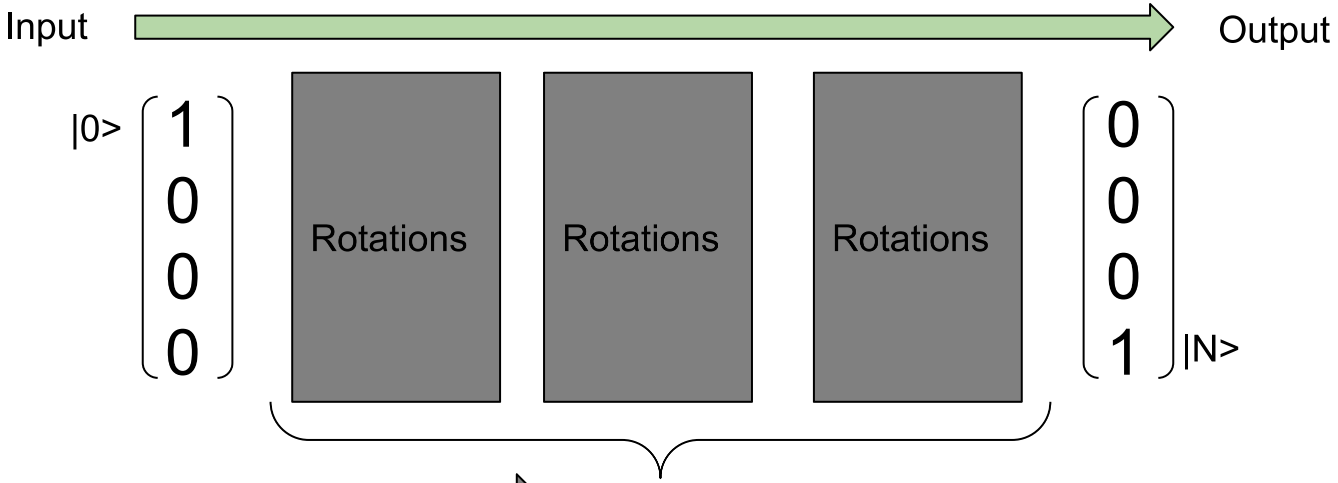
Use large values of M to create very small angles **theta = 1/M**
Example: sin(1/256) = 1/256, cos(1/256) = 1

k = approx. M

The difference between classical and quantum is the value of M!

Input to Output

The why: This is a sketch of a quantum circuit looks like



blue is for the Grover speedup

M times

Classical computer:

- exhaustive **$M=N=2^n$**
- random **$M=N/2=2^{n-1}$**

red is for classical runtime

Building rotations - Outer product - Rotations

The why: Mirroring against the two vectors has to be implemented mathematically

1) Transform **|0> to |1>** and **|1> to |0>** where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

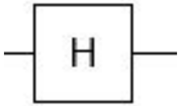
The **bit flip matrix X** = $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$ = **|0X1| + |1X0|**

2) Define a matrix that takes **|0> to |+> = |0>+|1> and |1> to |-> = |0>-|1>**

$|0\rangle(\langle 0| + \langle 1|) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$

$|1\rangle(\langle 1| - \langle 1|) = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}$

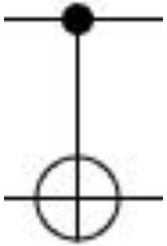
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ (almost) **Hadamard matrix**



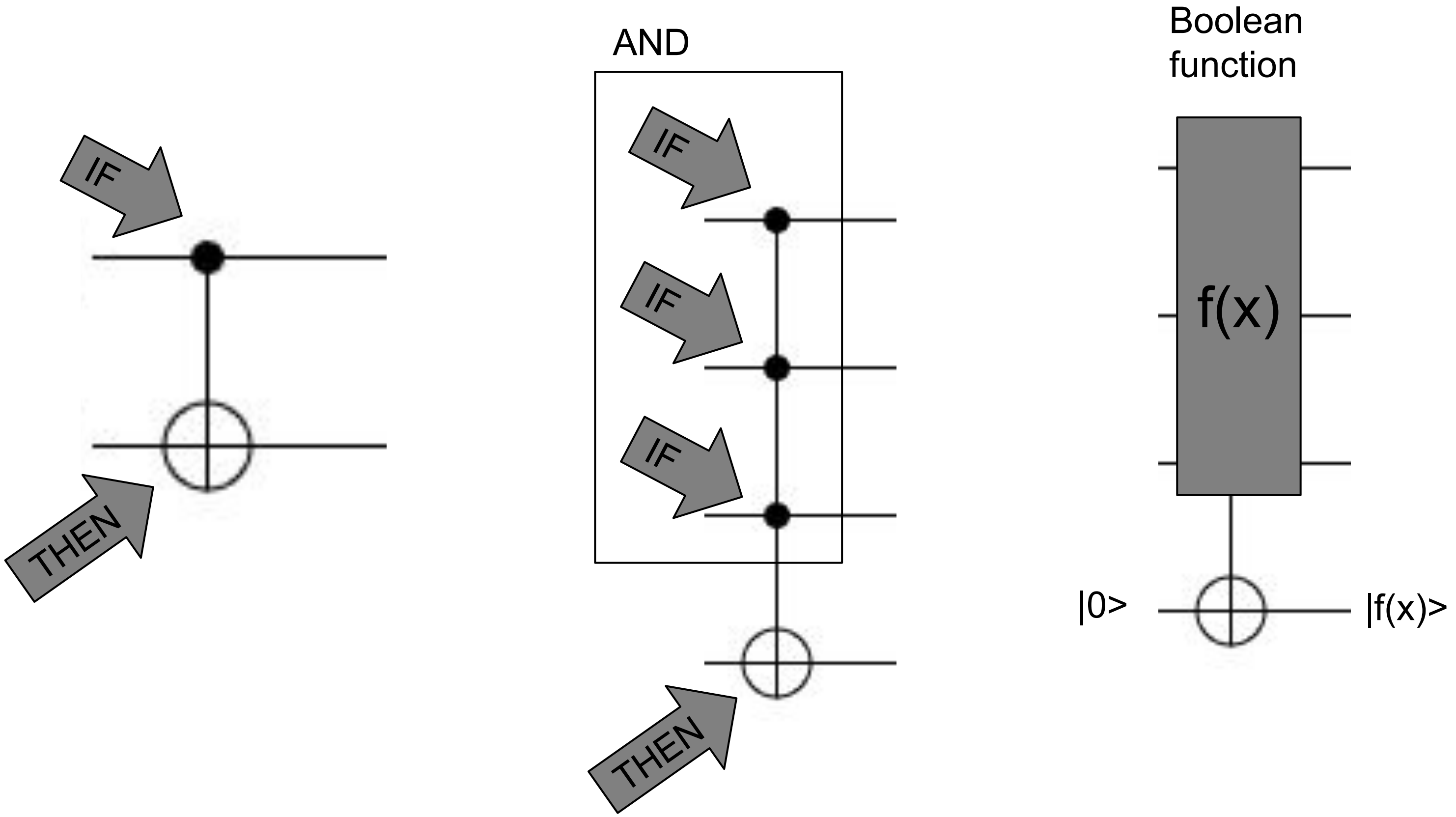
3) Define a matrix that **applies the X matrix only if the state of another vector is |1>**

|00X00| + |01X01| + |10X11| + |11X10|

$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ **CNOT matrix**



Building rotations - Encoding conditions for rotations



The why: Recipe for building rotations depending on configurable criteria

Classical problems can be imported into a quantum algorithm

The why: There is an exponential representational explosion that is often mentioned when quantum computations are discussed

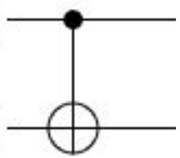
The quantum state

Previous statement: *All vectors have unit length*

A quantum state is a complex vector whose **L2 norm** is 1

- A qubit is a 2-dimensional complex vector. Examples |0>, |1>, |+>, |->
- The state of a n-qubit circuit is a 2^n-dimensional complex vector

  Example n=2, the state has four entries and the matrix has size 4 x 4

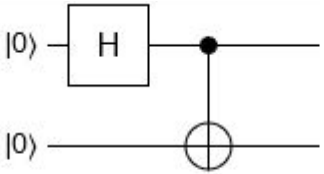


$$\underbrace{\left.\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\right\}2^n}_{2^n}$$

A quantum circuit is a 2^n x 2^n matrix

Entries in a state vector can be different from zero

Bell state $2^{-1/2}$(|00> + |11>)



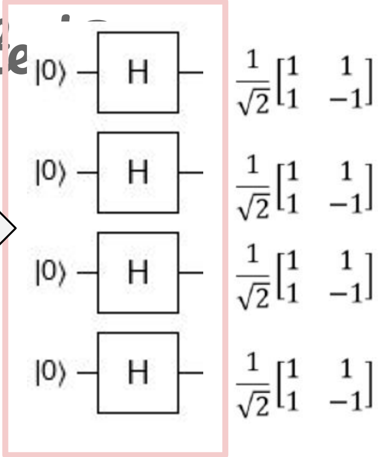
The superposition state

The why: We create a small enough angle necessary to implement the sequence of rotations with the necessary speedup

An n-qubit state has length 2^n

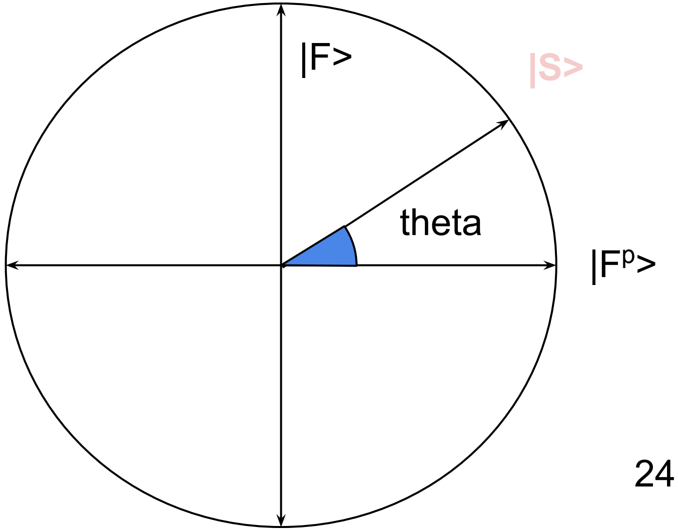
Define the **n-qubit** equal superposition |S> with H gates

$|S\rangle = \mathbf{2^{-n/2}}\,(|00....00\rangle + |00....01\rangle + \ldots + \boldsymbol{|11....11\rangle})$



Assume that the sought element is **|N>=|11....11>**

$$\langle F|S\rangle = \mathbf{2^{-n/2}} = \mathbf{1/M}$$

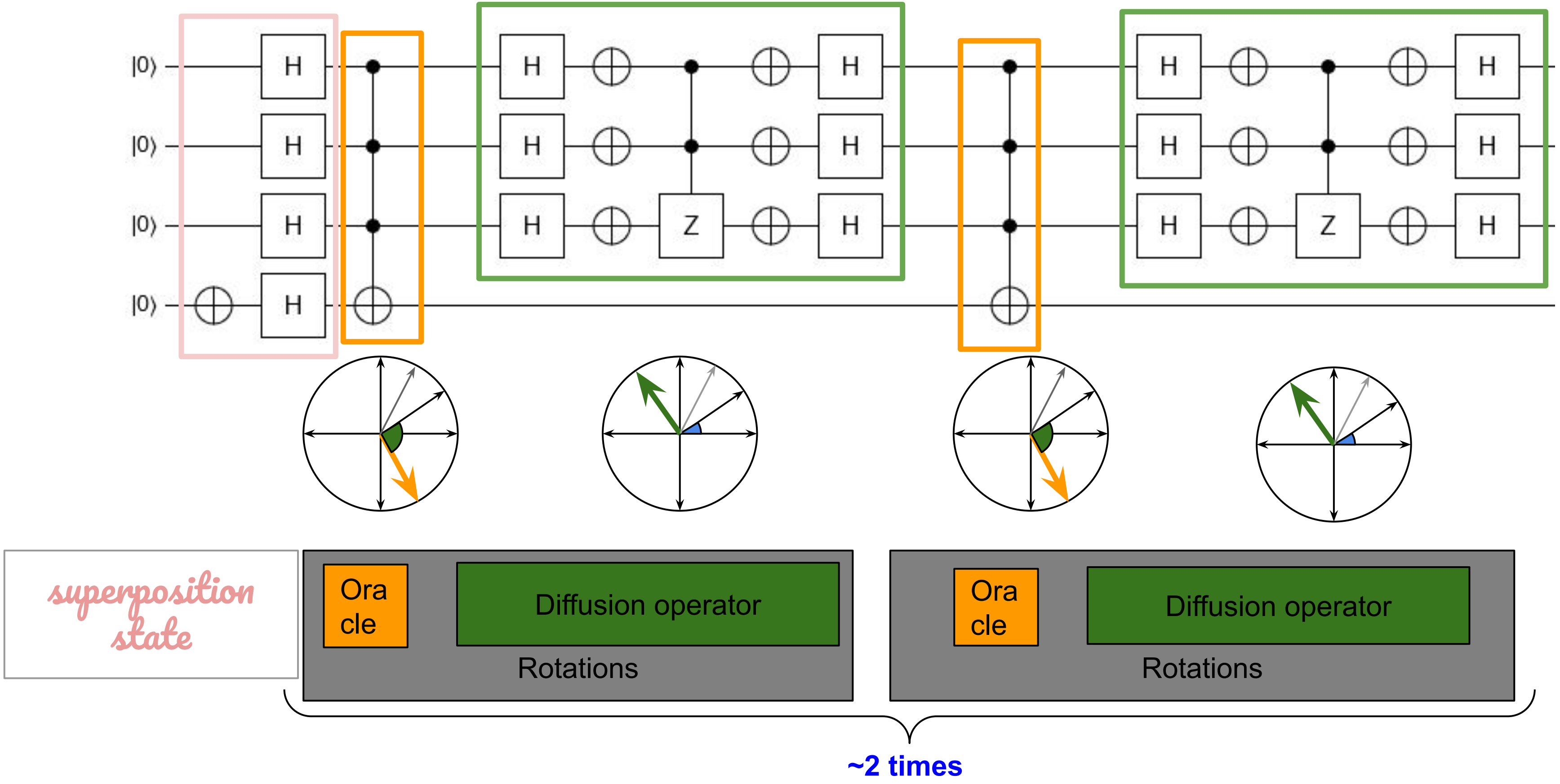


As a result, **M = sqrt(2^n) rotations are needed**

Each rotation (called Grover iteration) consists of

1) **mirror around $|F^p\rangle$**
2) **mirror around |S>**

The Grover search circuit for n=3 qubits



Grover's Algorithm Summary

For N = 1000 entries

- classical exhaustive search method needs 1000 steps
- Grover's algorithm needs approx. 32 steps

The key concepts presented:

- quantum qubit, gate, circuit
- how to import classical problems (Boolean logic) into quantum circuits

The key elements of the algorithm are:

- Mirroring operations
    - a known vector - the equal superposition state
    - a configurable vector - the search criteria
    - mirror operations are implemented with quantum gates
- The speed-up is from the L2 norm to calculate the distance between two qubit states