Chapter 4

Computational Examples

Some classical quantum algorithms will be re-examined, first mathematically, then in a quantum matter, to exemplify both the theory and the implementation.

- Deutsch problem
- Bernstein-Vazirani problem
- Grover algorithm
- Simon problem
- Constructing the Toffoli gate

4.1 Deutsch Problem

- Deutsch problem is a completely pointless problem.
- However, it is a perfect illustration of all that is miraculous, subtle, and disappointing about quantum computers.
  - ◇ It calculates a solution to a problem faster than any classical computer ever can.
  - ◇ It illustrates the subtle interaction of superposition, phase-kick back, and interference.

Problem Statement

- Suppose $f : \{0,1\} \to \{0,1\}$.

	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

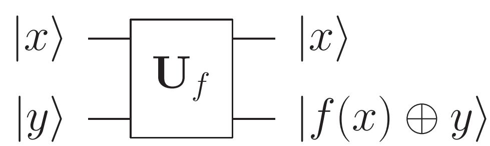
  Determine whether f is constant or balanced.

- The problem is "trivial", since we only need to evaluate f at $x = 0$ and $x = 1$.
  - ◇ Can we reach the conclusion by just one query?
  - ◇ The question boils down to evaluating $f(0) + f(1)$ by one query.

- Using (4.1), define the quantum computation of f via

$$\mathbf{U}_f(|x\rangle\,|y\rangle) = |x\rangle\,|f(x) \oplus y\rangle\,. \tag{4.1}$$

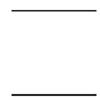
  and consider the general gate



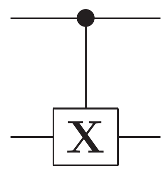
Gate Representation of f

- We can represent the four individual functions by the circuits:

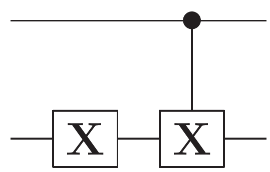
$\mathbf{U}_{f_0} =$



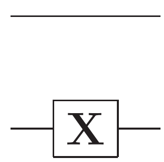
$\mathbf{U}_{f_1} =$



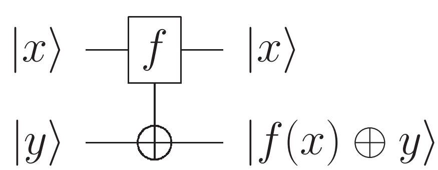
$\mathbf{U}_{f_2} =$



$\mathbf{U}_{f_3} =$



- We can also represent the gate $\mathbf{U}_f$ as a controlled-f operation

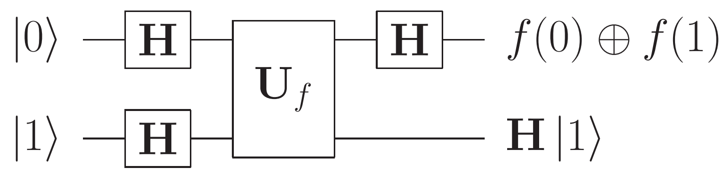


  which is called the Deutsch-Josza oracle.

- To be effective, we need to construct a quantum way to evaluate $f(0) + f(1)$.

Deutsch Algorithm

- Consider the circuit



  ◇ Prior to entering the black box $\mathbf{U}_f$, we have prepared the 2-qubit state

$$|0\rangle \otimes |1\rangle \Rrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

  ◇ The black box $\mathbf{U}_f$ does the following:

$$\mathbf{U}_f(\mathbf{H}|0\rangle\,\mathbf{H}|1\rangle) = \frac{1}{2}\mathbf{U}_f(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle))$$

$$= \frac{1}{2}((-1)^{f(0)}(|0\rangle(|0\rangle - |1\rangle)) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle))$$

$$= \begin{cases} (-1)^{f(0)}(|\mathbf{H}|0\rangle\rangle\,|\mathbf{H}|1\rangle\rangle), & \text{if } f(0) = f(1), \\ (-1)^{f(0)}(|\mathbf{H}|1\rangle\rangle\,|\mathbf{H}|1\rangle\rangle), & \text{if } f(0) \neq f(1). \end{cases}$$

  ◇ The final $\mathbf{H} \otimes I_2$ returns

$$\begin{cases} (-1)^{f(0)}(|0\rangle\,|\mathbf{H}|1\rangle\rangle), & \text{if } f(0) = f(1), \\ (-1)^{f(0)}(|1\rangle\,|\mathbf{H}|1\rangle\rangle), & \text{if } f(0) \neq f(1) \end{cases}$$

$$= (-1)^{f(0)}(|f(0) \oplus f(1)\rangle\,|\mathbf{H}|1\rangle\rangle).$$

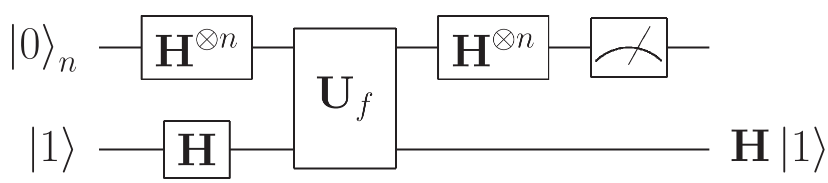
- The idea is about superposition, entanglement and interference.

  ◇ By measuring the input (top) register, we can indeed answer the Deutsch problem.

  ◇ The output register contains no useful information at all.

Deutsch-Jozsa algorithm

- Suppose $f : \{0,1\}^n \to \{0,1\}$ is either constant or is balanced. Determine which of the two is true.
  - ◇ If $n = 2$, then there are 2 constant functions and 6 balanced functions.
  - ◇ Elements in $\{0,1\}^n$ can be identified as $\{|x\rangle_n\}$.
- On a classical machine, we need to make $2^{n-1}+1$ queries.
- The Deutsch-Jozsa algorithm answer this question by just one query.



  - ◇ What is to be entered into $\mathbf{U}_f$?

$$|0\rangle_n \otimes |1\rangle \Rrightarrow \sum_{0\le x<2^n} \frac{1}{\sqrt{2^n}}\,|x\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

  - ◇ What is obtained out of the box $\mathbf{U}_f$?

$$\sum_{0\le x<2^n} \frac{(-1)^{f(x)}}{\sqrt{2^n}}\,|x\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

  - ◇ Using (3.12), the final return is

$$\sum_{0\le z<2^n}\sum_{0\le x<2^n} \frac{(-1)^{f(x)+x\cdot z}}{2^n}\,|z\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \qquad (4.2)$$

How to Interpret?

- The probability of the state $|0\rangle_n$ is given by

$$|\sum_{0\leq x<2^n} \frac{(-1)^{f(x)}}{2^n}|^2.$$

- If $f(x)$ is constant, then the probability for the state $|0\rangle_n$ is precisely 1. That is, the measurement must be 0.
- If $f(x)$ is balanced, then half of the x will produce $f(x) = 0$ and the other half produces $f(x) = 1$, making the probability of the state $|0\rangle_n$ perfectly and destructively interfered to 0.
  - ◇ Any measurement that is not $|0\rangle_n$ implies that f is not constant.

4.2 Bernstein-Vazirani Problem

- This is another artificial problem.
- The significance lies not in the intrinsic arithmetical interest of the problem, but in the fact that it can be solved dramatically and unambiguously faster on a quantum computer.

Problem Statement

- Suppose $f : \{0, 1\}^{\otimes n} \to \{0, 1\}$ is defined via

$$f(x) = a \cdot x = a_{n-1}x_{n-1} \oplus \ldots \oplus a_0x_0.$$

- Suppose that we have a way to evaluate $f(x)$. Find $|a\rangle_n$ with the smallest number of evaluations of f.
- On a classical machine,
  - ◇ Can take $x = 2^k$, $0 \leq k < n$, then $f(2^k) = a_k$.
  - ◇ Need a total of n evaluations.
- On a quantum machine, regardless the size of n, just need one invocation.

Algorithm

- Apply the Deutsch-Jozsa algorithm to $f(x) = a \cdot x$. By (4.2), we have

$$\sum_{0 \le z < 2^n} \sum_{0 \le x < 2^n} \frac{(-1)^{a \cdot x + x \cdot z}}{2^n} |z\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{4.3}$$

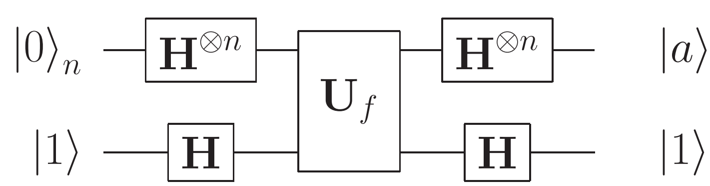
- For a fixed z, take a close look at the second summation:

$$\frac{1}{2^n} \sum_{0 \le x < 2^n} (-1)^{(a+z) \cdot x} = \frac{1}{2^n} \prod_{j=0}^{n-1} ((-1)^0 + (-1)^{a_j \oplus z_j}).$$

  ◇ If there exists one $0 \le j < n - 1$ at which $a_j \oplus z_j \neq 0$, the product is zero.

  ◇ If the coefficient associated with $|z\rangle_n$ is not zero, then it must that the $z \equiv a$ and that the associate coefficient must be 1.

- Thus, we can modify the Deutsch-Jozsa algorithm to



  ◇ Observe all n bits of the number a can now be determined by measuring the input register, whereas we have called the subroutine only once!

4.3 Simon Problem

- In the Bernstein-Vazirani problem,
  - ◇ A classical computer must call the subroutine n times to determine the value of a. The number of calls grows linearly with n.
  - ◇ A quantum computer need call the subroutine only once. The number of calls is independent of n.
- The Simon problem illustrates that the speed-up with a quantum computer can be substantially more dramatic.
  - ◇ With a classical computer the number of calls grows exponentially in n
  - ◇ With a quantum computer, the calls grow only linearly.

Problem Statement

- Suppose $F : \{0,1\}^n \to \{0,1\}^n$ is such that
  - ◇ F is periodic; namely, there exists $0 \neq p \in \{0,1\}^n$ such that $F(x \oplus p) = F(x)$ for every $x \in \{0,1\}^n$.
  - ◇ If $y \neq x$ and $F(x) = F(y)$, then $y = x \oplus p$.
- Find the period p.

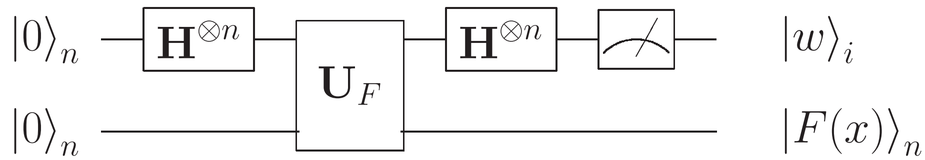
Conventional Search

- Algorithm:
  - ◇ Feed the function F with a sequence of different $x_1, x_2, x_3, \ldots$.
  - ◇ List the resulting values of F until we stumble on an $F(x_j)$ that is the same as the previously computed values $F(x_i)$.
  - ◇ Then $p = x_j \oplus x_i$.
- Complexity analysis:
  - ◇ At any stage of the process prior to the first success, if m different values of x have been tried, then all we know is that $p \neq x_i \oplus x_j$ for all pairs of previously selected values of x.
    - ▷ Thus at the mth state, only $\frac{m(m-1)}{2}$ candidates of p are eliminated.
  - ◇ There are a total of $2^n - 1$ possibilities for p.
  - ◇ To have probability ε of success after m trial, we need

$$1 - (1 - \frac{1}{2^n - 1})^{\frac{m(m-1)}{2}} \geq \varepsilon.$$

    - ▷ The number m of calls for achieving an appreciable probability of determining p grows exponentially in n.
    - ▷ Suppose $n = 100$. To have $\varepsilon = 50\%$ chance of success, we need to have tried approximately $m = 1.3256 \times 10^{15}$ calls.

Algorithm



- The black box function $\mathbf{U}_F$ is a bitwise generalization of $\mathbf{U}_f$ defined in (4.1), i.e.,

$$\mathbf{U}_F(|x\rangle_n |y\rangle_n) = |x\rangle_n |F(x) \oplus y\rangle_n . \tag{4.4}$$

- After the $\mathbf{U}_F$ operation, the state is

$$\mathbf{U}_F((\mathbf{H}^{\otimes n} \otimes I_n) |0\rangle_n |0\rangle_n) = \frac{1}{\sqrt{2^n}} \sum_{0 \le x < 2^n} |x\rangle_n |F(x)\rangle_n . \tag{4.5}$$

- Given $x \in \{0,1\}^n$, observe

$$\mathbf{H}^{\otimes n}(\frac{1}{\sqrt{2}} |x\rangle_n + \frac{1}{\sqrt{2}} |x \oplus p\rangle_n) = \frac{1}{\sqrt{2^{n-1}}} \sum_{z \perp p} (-1)^{x \cdot z} |z\rangle_n . \tag{4.6}$$

  ◇ Already seen in (3.12),

$$\mathbf{H}^{\otimes n} |x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{0 \le z < 2^n} (-1)^{x \cdot z} |z\rangle_n .$$

  ◇ Therefore,

$$\begin{aligned} \mathbf{H}^{\otimes n}(|x\rangle_n + |x \oplus p\rangle_n) &= \frac{1}{\sqrt{2^n}} \sum_{0 \le z < 2^n} (-1)^{x \cdot z}(1 + (-1)^{p \cdot z}) |z\rangle_n \\ &= \frac{2}{\sqrt{2^n}} \sum_{z \perp p} (-1)^{x \cdot z} |z\rangle_n . \end{aligned}$$

Interpretation

- The set $\{0,1\}^n$ can be partitioned into 2^{n-1} pairs of strings of the form $\{x, x \oplus p\}$.
  - ◇ Let $\mathscr{I}$ denote the subset consisting of one representative from each pair.
- The last application of $\mathbf{H}^{\otimes n}$ to (4.5) produces

$$
\begin{aligned}
&(\mathbf{H}^{\otimes n} \otimes I_n)(\frac{1}{\sqrt{2^n}} \sum_{0 \le x < 2^n} |x\rangle_n |F(x)\rangle_n) \\
= &\frac{1}{\sqrt{2^{n-1}}} \sum_{x \in \mathscr{I}} \mathbf{H}^{\otimes n}(\frac{1}{\sqrt{2}} |x\rangle_n + \frac{1}{\sqrt{2}} |x \oplus p\rangle_n) |F(x)\rangle_n \\
= &\frac{1}{\sqrt{2^{n-1}}} \sum_{x \in \mathscr{I}} (\frac{1}{\sqrt{2^{n-1}}} \sum_{z \perp p} (-1)^{x \cdot z} |z\rangle_n) |F(x)\rangle_n
\end{aligned}
$$

- The first register is an equally weighted superposition of elements $|z\rangle_n$ that are perpendicular to p.
  - ◇ Measure the first register and we learn one value $|z_i\rangle_n$ satisfying $z_i \perp p$.
  - ◇ If the dimension of the subspace span$\{z_1, \ldots, z_m\}$ is less than $n-1$, rerun the circuit until there is enough vectors.
    - ▷ Solve the linear equation $Zp = 0$ for a nontrivial p.
  - ◇ Each new z_i eliminates half of the candidates for p.
    - ▷ If $m = n + t$, then the probability of determining p is at least $1 - \frac{1}{2^{t+1}}$.
    - ▷ With $O(n)$ trials, there is a good probability of find p.

4.4 Unstructured Search Problem

- The quantum search algorithm performs a generic search for a solution through a space of potential solutions.
- The extremely wide applicability of searching problems makes Grovers algorithm interesting and important.
- The focus is at the polynomial speed-up over the best-known classical algorithms.

Problem Statement

- Given a black box function $f : \{0,1\}^n \rightarrow \{0,1\}$ and being promised that there is a unique $a \in \{0,1\}^n$ such that $f(a) = 1$, find a.
- The underlying search is called "unstructured" because we have no prior knowledge about the contents of the database.
  - ◇ Unstructured search can be thought of as a database search problem in which we want to find an item that meets some specification.
  - ◇ If there is a way to "sort" the database, then we might perform binary search in logarithmic time.
- On a classical machine, we have no way but check the items one by one and it will take $O(2^n)$ steps on average.
- The Grover's algorithm takes only $O(2^{\frac{n}{2}})$ steps.
  - ◇ This is accomplished by amplifying the amplitude of the vector $|a\rangle$ while canceling those of the vectors $|x\rangle$ for $x \neq a$.
  - ◇ Equivalently, the quantum algorithm is said to have provided a quadratic speed-up over classical exhaustive search.

Grover Iterate

- Recall the fact that

$$\mathbf{U}_f \left|x\right\rangle_n \left|\mathbf{H}\left|1\right\rangle\right\rangle = (-1)^{f(x)} \left|x\right\rangle_n \left|\mathbf{H}\left|1\right\rangle\right\rangle.$$

  ◇ We abbreviate this special phase transformation as

$$\mathbf{U}_f \left|x\right\rangle_n = (-1)^{f(x)} \left|x\right\rangle_n.$$

- If $\left|\Psi\right\rangle = \sum_x \omega_x \left|x\right\rangle_n$, then

$$\begin{aligned} \mathbf{U}_f \left|\Psi\right\rangle &= \sum_x (-1)^{f(x)} \omega_x \left|x\right\rangle_n = \sum_x \omega_x \left|x\right\rangle - 2\omega_a \left|a\right\rangle_n \\ &= \left(I_n - 2\left|a\right\rangle_n \left\langle a\right|_n\right) \left|\Psi\right\rangle. \end{aligned}$$

  ◇ Can identify $\mathbf{U}_f = I_n - 2\left|a\right\rangle \left\langle a\right|$ without knowing a.

  ◇ $\mathbf{U}_f$ flips the sign of the component associated with $\left|a\right\rangle$, but leaves others unchanged.

  ◇ In linear algebra, $\mathbf{U}_f$ acts like the Householder reflector.

- For $\left|\phi\right\rangle := \mathbf{H}^{\otimes n} \left|0\right\rangle_n = \sum_x \frac{1}{\sqrt{2^n}} \left|x\right\rangle_n$, define

$$W := 2\left|\phi\right\rangle \left\langle\phi\right| - I_n. \tag{4.7}$$

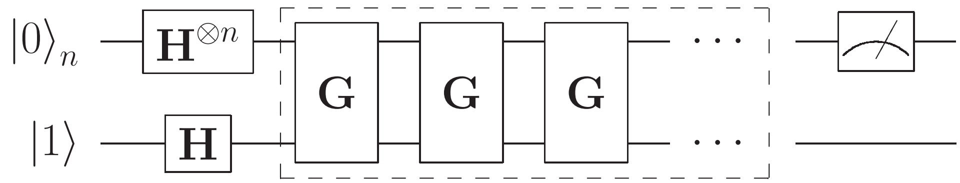
  ◇ Observe that

$$W\mathbf{H}^{\otimes n} \left|x\right\rangle = \begin{cases} \left|\phi\right\rangle, & \text{if } \left|x\right\rangle = \left|0\right\rangle, \\ -\mathbf{H}^{\otimes n} \left|x\right\rangle, & \text{if } \left|x\right\rangle \neq \left|0\right\rangle. \end{cases}$$

- The operator $G := W\mathbf{U}_f \otimes I_2$ is called a *Grover iterate*.

Grover Algorithm

- The Grover algorithm applies the iterate G about $\frac{\pi}{4}\sqrt{2^n}$ times and take the measurement.



- One-step analysis:
  - ◇ Before the first G application,

$$|0\rangle_n |1\rangle \Rrightarrow \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \, \mathbf{H} \, |1\rangle \, .$$

  - ◇ After $\mathbf{U}_f$,

$$\Rrightarrow \frac{1}{\sqrt{2^n}} (\sum_x |x\rangle - 2\,|a\rangle) \mathbf{H} \, |1\rangle \, .$$

  - ◇ After W,

$$\begin{aligned} \Rrightarrow \quad & ((1 - \frac{4}{2^n}) \, |\phi\rangle + \frac{2}{\sqrt{2^n}} \, |a\rangle) \mathbf{H} \, |1\rangle \\ & = ((\frac{1}{\sqrt{2^n}} - \frac{4}{\sqrt{2^{3n}}}) \sum_{x \neq a} |x\rangle + (\frac{3}{\sqrt{2^n}} - \frac{4}{\sqrt{2^{3n}}}) \, |a\rangle) \mathbf{H} \, |1\rangle \, . \end{aligned}$$

  - ◇ The main point of such a G application is that the probability of $|a\rangle$ is slightly increased. (Check to see that the total probability is still added to one.)

Basic Mechanism

- Use these two mechanism in the Grover iterate:

$$\left\{ \begin{array}{rcl} \mathbf{U}_f \left| a \right\rangle & = & -\left| a \right\rangle , \\ \mathbf{U}_f \left| \phi \right\rangle & = & \left| \phi \right\rangle - \frac{2}{\sqrt{2^n}} \left| a \right\rangle . \end{array} \right. \tag{4.8}$$

$$\left\{ \begin{array}{rcl} W \left| \phi \right\rangle & = & \left| \phi \right\rangle , \\ W \left| a \right\rangle & = & \frac{2}{\sqrt{2^n}} \left| \phi \right\rangle - \left| a \right\rangle . \end{array} \right. \tag{4.9}$$

- Apply $\mathbf{G}$ gate repeatedly to the general form

$$\left| \Psi \right\rangle = s \left| \phi \right\rangle + t \left| a \right\rangle$$

with $(\frac{s}{\sqrt{2^n}})^2(2^n - 1) + (\frac{s}{\sqrt{2^n}} + t)^2 = 1$.

- Therefore,

$$\begin{aligned} W\mathbf{U}_f \left| \Psi \right\rangle &= W(s\mathbf{U}_f \left| \phi \right\rangle + t\mathbf{U}_f \left| a \right\rangle) \\ &= W(s(\left| \phi \right\rangle - \frac{2}{\sqrt{2^n}} \left| a \right\rangle) - t \left| a \right\rangle) \\ &= sW \left| \phi \right\rangle - (t + \frac{2s}{\sqrt{2^n}})W \left| a \right\rangle \\ &= s \left| \phi \right\rangle - (t + \frac{2s}{\sqrt{2^n}})(\frac{2}{\sqrt{2^n}} \left| \phi \right\rangle - \left| a \right\rangle) \\ &= (s - \frac{4s}{2^n} - \frac{2t}{\sqrt{2^n}}) \left| \phi \right\rangle + (t + \frac{2s}{\sqrt{2^n}}) \left| a \right\rangle . \end{aligned}$$

  - ◇ The probability for $\left| a \right\rangle$ is given by

$$\left\{ \begin{array}{rl} (\frac{s}{\sqrt{2^n}} + t)^2, & \text{before } \mathbf{G} \ , \\ (\frac{3s}{\sqrt{2^n}} + t - \frac{t}{2^{n-1}} - \frac{4s}{\sqrt{2^{3n}}})^2, & \text{after } \mathbf{G} \ . \end{array} \right. \tag{4.10}$$

Dynamics of Probabilities

- Given $s_0 = 1$ and $t_0 = 0$, the Grover algorithm generates a sequence of coefficients

$$\begin{cases} s_{k+1} & := & s_k - \frac{4s_k}{2^n} - \frac{2t_k}{\sqrt{2^n}}, \\ t_{k+1} & := & t_k + \frac{2s_k}{\sqrt{2^n}} \end{cases} \tag{4.11}$$

  - ◇ Show that the sequence (s_k, t_k) satisfies the relationship $s^2 + 2\frac{st}{\sqrt{N}} + t^2 = 1$, which is an ellipse.
  - ◇ Rewrite the iteration in matrix form

$$\begin{bmatrix} s_{k+1} \\ t_{k+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{4}{2^n} & -\frac{2}{\sqrt{2^n}} \\ \frac{2}{\sqrt{2^n}} & 1 \end{bmatrix} \begin{bmatrix} s_k \\ t_k \end{bmatrix}. \tag{4.12}$$

    - ▷ Eigenvalues $\frac{2^n - 2 \pm 2\sqrt{1-2^n}}{2^n}$ are complex and have moduli 1.
    - ▷ The iterates (s_k, t_k) will not converge (cycle around the ellipse).

- Find the first k that will maximize the probability for $|a\rangle$.
  - ◇ Do not iterate more than $O(2^{n-1})$ times. (Why?)

4.5 Constructing the Toffoli gate

- For a reversible classical computer, it can be shown that at least one 3-bit gate, such as the **ccNOT**, is needed to build up general logical operations.
  - ◇ It is also known that such 3-bit gates cannot be built up out of 1- and 2-bit gates.
- In a quantum computer, however, it is remarkable and importantl for the feasibility of practical quantum computation that the quantum extension of this 3-qbit gate, such as the Toffoli gate **T**, can be constructed out of a small number of 1- and 2-qbit gates.
- We will come back to work on this section later.